Cincom

AD/ADVANTAGE

MANTIS Quick Reference
OS/390, VSE/ESA

P39-5003-00

AD/Advantage® MANTIS Quick Reference OS/390, VSE/ESA

Publication Number P39-5003-00

Cincom Systems, Inc.
55 Merchant Street
Cincinnati, OH 45246-3732

PHONE: (513) 612-2300
FAX: (513) 612-2000
WORLD WIDE WEB: http://www.cincom.com

Attention:

Some Cincom products, programs, or services referred to in this publication may not be available in all countries in which Cincom does business. Additionally, some Cincom products, programs, or services may not be available for all operating systems or all product releases. Contact your Cincom representative to be certain the items are available to you.

Release information for this manual

AD/Advantage MANTIS Quick Reference, OS/390, VSE/ESA, P39-5003-00, is dated October 30, 2001. This document supports Release 5.5.01 of MANTIS.

We welcome your comments

We encourage critiques concerning the technical content and organization of this manual. Please take the survey provided with the online documentation at your convenience.

Cincom Technical Support for AD/Advantage

All customers	Web:	http://supportweb.cincom.com
U. S. A. customers	Phone:	1-800-727-3525
	FAX:	(513) 612-2000 Attn: AD/Advantage Support
	Mail:	Cincom Systems, Inc. Attn: AD/Advantage Support 55 Merchant Street Cincinnati, OH 45246-3732 U. S. A.
Customers outside U. S. A.	All:	Visit the support links at http://www.cincom.com to find contact information for your nearest Customer Service Center.

Contents

About this book xiii

Program Design Facility commands 21

About this book

Using this document

This quick reference is intended to provide a brief overview of the commands and functions for MANTIS for the IBM® mainframe environment.

Document organization

Chapter 1—Program Design Facility commands

For each of the Program Design Facility commands, provides a description, syntax illustration, and example .

Chapter 2—Full-Screen Editor (FSE) commands

Contains a table stating whether each FSE command is a line command or a primary command. Then, for each FSE command, provides a description, syntax illustration, and example.

Chapter 3—MANTIS statements, functions, and commands

For each of the MANTIS statements, functions, and commands, provides a description, syntax illustration, and example.

Chapter 4—Built-in functions and constants

Contains a table that lists the following for each built-in function or constant: description, input, output, and function type.

Chapter 5—Special characters

Contains a table that describes each MANTIS special character.

Chapter 6—Operators

List MANTIS operators in order of evaluation.

Chapter 7—Attribute types

Lists, by type, attribute values that you can set in attribute statements or in screen design.

Chapter 8—ATTRIBUTE function (values returned)

Lists the values that can be returned by the ATTRIBUTE function. (Underlining indicates the abbreviation that MANTIS returns.)

Chapter 9—Screen Design PF keys

Provides a table describing Screen Design PF keys used to create or update a screen, update field specifications or repeat specifications, and affect the screen display.

Chapter 10—Screen Design commands

Provides a table describing Screen Design commands.

Chapter 11—Search Facility

Describes how to access the MANTIS Search Facility and describes all of the fields on the MANTIS Search Facility screen.

Chapter 12—Universal Export Facility (UEF)

Describes all of the fields on the UEF screen and describes UEF syntax.

Revisions to this manual

The following revisions are for Release 5.5.01:

- We have added the "LUID Function" section on page 82.
- We have added the LUID function added to the table beginning on page 115.
- We have added the following chapters:
  - "Search Facility" on page 137
  - "Universal Export Facility (UEF)" on page 143

Conventions

The following table describes the conventions used in this document series:

Convention	Description	Example

`Constant width type`	Represents screen images and segments of code.	`Screen Design Facility`
		`GET NAME LAST`
		`INSERT ADDRESS`

Yellow-highlighted, red code or screen text	Indicates an emphasized section of code or portion of a screen.	`00010 ENTRY COMPOUND`
		`00020 .SHOW"WHAT IS THE`
		`  CAPITAL AMOUNT?"`
		`00030 .OBTAIN INVESTMENT`
		`00040 EXIT`

Slashed b (b̸)	Indicates a space (blank). The example indicates that a password can have a trailing blank.	`WRITEPASSb̸`
Brackets [ ]	Indicate optional selection of parameters. (Do not attempt to enter brackets or to stack parameters.) Brackets indicate one of the following situations.	
	A single item enclosed by brackets indicates that the item is optional and can be omitted. The example indicates that you can optionally enter a program name.	COMPOSE [*program-name*]
	Stacked items enclosed by brackets represent optional alternatives, one of which can be selected. The example indicates that you can optionally enter NEXT, PRIOR, FIRST, or LAST. (NEXT is underlined to indicate that it is the default.)	**NEXT** **PRIOR** **FIRST** **LAST**

Convention	Description	Example
Braces { }	Indicate selection of parameters. (Do not attempt to enter braces or to stack parameters.) Braces surrounding stacked items represent alternatives, one of which you must select. The example indicates that you must enter FIRST, LAST, or a value for *begin*.	$\left\{\begin{array}{l}\textbf{FIRST}\\ \textbf{\textit{begin}}\\ \textbf{LAST}\end{array}\right\}$
$\underline{\text{Underlining}}$ F(In syntax)	Indicates the default value supplied when you omit a parameter. The example indicates that if you do not specify ON, OFF, or a row and column destination, the system defaults to ON.	$\textbf{SCROLL}\left[\begin{array}{l}\underline{\textbf{ON}}\\ \textbf{OFF}\\ [\textbf{\textit{row}}][\textbf{\textit{,col}}]\end{array}\right]$
	Underlining also indicates an allowable abbreviation or the shortest truncation allowed. The example indicates that you can enter either PRO or PROTECTED.	$\underline{\texttt{PRO}}\texttt{TECTED}$
Ellipsis points...	Indicate that the preceding item can be repeated. The example indicates that you can enter (A), (A,B), (A,B,C), or some other argument in the same pattern.	`(`*`argument`*`,...)`

Convention	Description	Example
UPPERCASE	Indicates MANTIS reserved words. You must enter them exactly as they appear. The example indicates that you must enter CONVERSE exactly as it appears.	`CONVERSE` *`name`*
Italics	Indicate variables you replace with a value, a column name, a file name, and so on. The example indicates that you can supply a name for the program.	`COMPOSE [`*`program-name`*`]`
Punctuation marks	Indicate required syntax that you must code exactly as presented. ( ) parentheses . period , comma : colon ; semicolon ' single quotation mark " " double quotation marks	$[\textbf{LET}]v\begin{bmatrix}\textbf{(}i\textbf{)}\\ \textbf{(}i, j\textbf{)}\end{bmatrix}[\textbf{ROUNDED(}n\textbf{)}]\textbf{=}\,e1\ [,e2,e3\ldots]$

MANTIS documentation series

MANTIS is an application development system designed to increase productivity in all areas of application development, from initial design through production and maintenance. MANTIS is part of AD/Advantage, which offers additional tools for application development. Listed below are the manuals offered with MANTIS in the IBM® mainframe environment, organized by task. You may not have all the manuals listed here.

MASTER User tasks

- *MANTIS Installation, Startup, and Configuration, MVS/ESA*, *OS/390,* P39-5018
- *MANTIS Installation, Startup, and Configuration, VSE/ESA,* P39-5019
- *MANTIS Administration, OS/390, VSE/ESA*, P39-5005
- *MANTIS Messages and Codes, OS/390, VSE/ESA*, P39-5004*
- *MANTIS Administration Tutorial, OS/390, VSE/ESA,* P39-5027
- *MANTIS XREF Administration, OS/390, VSE/ESA*, P39-0012

General use

- *MANTIS Quick Reference, OS/390, VSE/ESA*, P39-5003
- *MANTIS Facilities, OS/390, VSE/ESA*, P39-5001
- *MANTIS Language, OS/390, VSE/ESA*, P39-5002
- *MANTIS Program Design and Editing, OS/390, VSE/ESA*, P39-5013
- *MANTIS Messages and Codes, OS/390, VSE/ESA*, P39-5004*
- *AD/Advantage Programming*, P39-7001
- *MANTIS DB2 Programming, OS/390, VSE/ESA*, P39-5028

- *MANTIS SUPRA SQL Programming, OS/390, VSE/ESA*, P39-3105
- *MANTIS XREF, OS/390, VSE/ESA, OpenVMS*, P39-0011
- *MANTIS Entity Transformers*, P39-0013
- *MANTIS DL/I Programming, OS/390, VSE/ESA*, P39-5008
- *MANTIS SAP Facility, OS/390, VSE/ESA*, P39-7000
- *MANTIS WebSphere MQ Programming*, P39-1365
- *MANTIS Application Development Tutorial, OS/390, VSE/ESA*, P39-5026

NOTE

Manuals marked with an asterisk (*) are listed twice because you use them for both MASTER User tasks and general use tasks.

Educational material

AD/Advantage and MANTIS educational material is available from your regional Cincom education department.

1

Program Design Facility commands

ACTION

Displays the action bar across the top of list panels.

$$\left\{\begin{array}{l}\textbf{ACTION}\\ \textbf{ACTN}\end{array}\right\}$$

Example

```
ACTION
```

AUDIT

Displays the Audit Trail List.

AUDIT

Example

```
AUDIT
```

BILL

Displays the Bill of Materials List.

BILL [*program - name*]

Example

```
BILL EXAMPLE_PROGRAM
```

BIND

Creates an HPO-bound version of a MANTIS program.

BIND [*program-name*]

Example

```
BIND EXAMPLE_PROGRAM
```

BROWSE

Lets you display profile information for a program in your directory.

BROWSE [*program_ name*]

Example

```
BROWSE EXAMPLE_PROGRAM
```

CANCEL

Allows you to exit from a session with the Program Design Facility, one panel at a time, or to exit from the action bar pull-down to the action bar.

CANCEL

Example

```
CANCEL
```

CEFCHECK

Identifies program components and source code that changed since the last Compose was issued on a source program.

CEFCHECK [*program - name*]

Example

```
CEFCHECK EXAMPLE_PROGRAM
```

CHECK

Checks an HPO-bound program to determine if any programs or components changed since the last time the program was bound.

CHECK [*program - name*]

Example

```
CHECK EXAMPLE_PROGRAM
```

CLEAR

Clears a panel of data. (Common dialog action that supports the 3270 hardware.)

CLEAR

Example

```
CLEAR
```

COMMAND

Toggles the command line from the top of a panel to the bottom (or from the bottom to the top).

$$\left\{\begin{matrix}\textbf{COMMAND}\\ \textbf{CMD}\end{matrix}\right\}$$

Example

```
COMMAND
```

COMPOSE

Assembles a source program and its COMPONENT statement(s) into an executable program with expanded component code.

COMPOSE [*program-name*]

Example

```
COMPOSE EXAMPLE_PROGRAM
```

CONFIRM

Confirms the execution of an action from a parameter entry panel.

CONFIRM

Example

```
CONFIRM
```

COPY

Copies the contents of a program from your library (or another library) to a program in your library.

COPY [*type*.] [$\left\{\begin{matrix}\text{"}\\ \text{'}\end{matrix}\right\}$ ***library*:] [*program - name*] [/*password*]** $\left[\begin{matrix}\text{"}\\ \text{'}\end{matrix}\right]$

Example

```
COPY PGRM. "ACCT:CUST_UPDATE"
COPY PGRM. 'ACCT:CUST_UPDATE'
```

CREF

Cross references programs and components in your library and then builds the Bill of Materials List from the cross reference.

CREF [*program - name*]

Example

```
CREF EXAMPLE_PROGRAM
```

DECOMPOSE

Disassembles an executable program into individual components and then updates program libraries with source changes and component changes.

DECOMPOSE [*program-name*]

Example

```
DECOMPOSE EXAMPLE_PROGRAM
```

DELETE

Deletes a record from the Trigger file.

DELETE [*type.*] [*sequence number*]

Example

```
DELETE ETRG. 0200
```

EDIT

Starts a session with the Full-Screen Editor (FSE) where you can view, create, and modify MANTIS programs.

EDIT [*type.*] [$\left\{\begin{matrix}\text{"}\\ \text{'}\end{matrix}\right\}$ ***library*:] [*program - name*] [/*password*] [/*description*]** $\begin{bmatrix}\text{"}\\ \text{'}\end{bmatrix}$

Example

```
EDIT PGRM. "ACCT:EXAMPLE_PROGRAM"
EDIT PGRM. 'ACCT:EXAMPLE_PROGRAM'
```

ET

Accesses the Entity Transformers facility directly from the Program Design Facility menu or program list, in addition to access from the MANTIS Facility Selection menu.

ET

Example

```
ET
```

EXECUTE

Executes an action on a trigger record or parameter entry panel, or updates the program profile.

EXECUTE [*type*.] [*sequence number*]

Example

```
EXECUTE ETRG. 2
```

EXHELP

Displays a help panel to explain a specific action.

EXHELP

Example

```
EXHELP
```

EXIT

Terminates the current function and returns to a higher level function.

EXIT

Example

```
EXIT
```

FORWARD

Repositions a list panel forward one panel, or retrieves the next record on a browse panel.

$$\left\{\begin{array}{l}\textbf{FORWARD}\\ \textbf{FWD}\end{array}\right\}$$

Example

```
FORWARD
```

HELP

Displays a help panel that explains a specific field, command, or message, or displays the KEYSTEMP panel where you can alter PF key settings.

$$\textbf{HELP}\left\{\begin{array}{l}\textbf{\textit{xxx}}\\ \textbf{\textit{command}}\\ \textbf{\textit{keys}}\end{array}\right\}$$

Example

```
    HELP COMPOSE
```

KEYSUPDATE

Displays a list of PF key settings that you can change for the duration of the current action.

$$\left\{\begin{array}{l}\textbf{KEYSUPDATE}\\ \textbf{KUPD}\end{array}\right\}$$

Example

```
KEYSUPDATE
```

L (LOCATE)

Repositions a list panel to a specific program or component.

L [*name*]

Example

```
L
```

LEFT

Moves the columns of a list panel the specified amount to the left to allow you to view all fields that extend beyond the width of your screen.

LEFT [*n*]

Example

```
LEFT 4
```

LIST

Displays the Program Directory List to let you select programs for editing with the Full-Screen Editor, or to let you select programs for other actions.

LIST [*type.*] [*program - name*]

Example

```
LIST EXAMPLE_PROGRAM
```

LOGOFF

Exits from MANTIS.

LOGOFF

Example

```
LOGOFF
```

MENU

Returns the MANTIS Facility Selection Menu.

MENU

Example

```
MENU
```

PROFILE

Displays program profile information.

PROFILE [*program-name*]

Example

```
PROFILE EXAMPLE_PROGRAM
```

PROMPT

Displays a list of the valid commands and actions you can issue from the current panel.

$$\left\{\begin{array}{l}\textbf{PROMPT}\\\textbf{PMPT}\end{array}\right\}$$

Example

```
PROMPT
```

PURGE

Deletes a program from your directory.

PURGE [*program - name*]

Example

```
PURGE EXAMPLE_PROGRAM
```

REFRESH

Updates date and time on list panels, restores Action fields on list panels, incorporates new entries on list panels, removes deleted entries from list panels, and resets Entry and Function Options on parameter entry panels.

REFRESH

Example

```
REFRESH
```

RENAME

Renames a program <u>from</u> your library <u>to</u> your library, and allows you to change program name, description, and password.

RENAME [*program - name*]

Example

```
RENAME EXAMPLE_PROGRAM
```

RETRIEVE

Redisplays the last seven commands, one at a time, that you issued from the command line of a panel.

$$\left\{\begin{matrix}\textbf{RETRIEVE}\\ \textbf{?}\end{matrix}\right\}$$

Example

```
RETRIEVE
```

RIGHT

Moves the columns of a list panel the specified amount to the right to allow you to view all fields that extend beyond the width of your screen.

RIGHT [*n*]

Example

```
RIGHT 6
```

SKIP

Bypasses execution of an action on a parameter entry panel for a specific program.

SKIP

Example

```
SKIP
```

SQLBIND

The SQLBIND command is for DB2 for VSE and VM (formerly SQL/DS) and DB2 only. You can use the SQLBIND command with static or extended dynamic SQL:

♦ **For static SQL.** Use the SQLBIND command to place information about a program's SQL statements and their host variables into an internal file, in order to create an SQL support module for static execution of the program.

♦ **For extended dynamic SQL.** Use the SQLBIND command to:

- Create a DB2 for VSE and VM access module for the program.
- Save information about SQL statements and host variables.
- Make the program immediately executable at the end of the bind.

SQLBIND [*program - name*]

Example

```
SQLBIND EXAMPLE_PROGRAM
```

SQLCHECK

The SQLCHECK command is for DB2 for VSE and VM (formerly SQL/DS) and DB2 only. You can use the SQLCHECK command with static or extended dynamic SQL:

♦ **For static SQL.** Use the SQLCHECK command to determine if a program and its corresponding SQL support load module are consistent.

♦ **For extended dynamic SQL.** Use the SQLCHECK command to determine if the program and its corresponding DB2 for VSE and VM access module are consistent.

SQLCHECK [*program - name*]

Example

```
SQLCHECK EXAMPLE_PROGRAM
```

SQLMAINT

The SQLMAINT command is for DB2 only. The SQLMAINT command allows you to view and/or purge SQL bind information. When you issue SQLMAINT, the SQL bind information panel is returned. This panel displays the program name, the corresponding module name and the date and time when the program was bound.

SQLMAINT

Example

```
SQLMAINT
```

SQLUNBIND

The SQLUNBIND command is for DB2 for VSE and VM (formerly SQL/DS) and DB2 only. You can use the SQLUNBIND command with static or extended dynamic SQL:

- **For static SQL.** Use the SQLUNBIND command to:
  - Mark a MANTIS program as not SQL-bound.
  - Delete the SQL bind information from the internal file.
- **For extended dynamic SQL.** Use the SQLUNBIND command to:
  - Mark the program as not SQL-bound.
  - Delete the associated DB2 for VSE and VM access module.

SQLUNBIND [*program - name*]

Example

```
SQLUNBIND EXAMPLE_PROGRAM
```

TRANSFER

Accesses the Transfer Facility directly from the Program Design Menu or Program List, in addition to Access from the MANTIS Facility Selection Menu.

TRANSFER

Example

```
TRANSFER
```

UNBIND

Replaces the HPO-bound version of a MANTIS program with the unbound version.

UNBIND [*program - name*]

Example

```
UNBIND EXAMPLE_PROGRAM
```

UPDATE

Updates program profile information and trigger records.

UPDATE [*type.*] [*sequence number*]

Example

```
UPDATE ETRG. 2
```

2

Full-Screen Editor (FSE) commands

The table below indicates whether the Full-Screen Editor (FSE) command is a line command or a primary command.

Command	Primary	Line
A (after)		✓
B (before)		✓
BIND	✓	
BOTTOM	✓	
C (copy)		✓
CANCEL	✓	
CHANGE	✓	
COPY	✓	
D (delete)		✓
DOWN	✓	
END	✓	
ERASE	✓	
ERRCODE	✓	
FIND	✓	
HELP	✓	
I (insert)		✓
KILL	✓	

Command	Primary	Line
LEFT	✓	
LIST	✓	
LOAD	✓	
LOCATE	✓	
LOGOFF	✓	
M (move)		✓
MENU	✓	
NEW	✓	
O (overlay)		
PRINT	✓	
PROFILE	✓	
PURGE	✓	
QUIT	✓	
R (repeat)		✓
RCHANGE	✓	
REPLACE	✓	
RESET	✓	
RFIND	✓	
RIGHT	✓	
RUN	✓	
S (select)		✓
SAVE	✓	
SCROLL	✓	
SEQUENCE	✓	
TOP	✓	
UP	✓	

A (after)

Line command used with the COPY or MOVE commands to indicate that you want the copied or moved line(s) to appear after the line marked with an “A”.

See also “B (before)” on page 39 and “O (overlay)” on page 49.

A

Example

```
a0040  .WHILE RECORD<>"END"
00050  .END
m0060  .CONVERSE MAP
00070  .STOP
00080  EXIT
```

B (before)

Line command used with the COPY or MOVE commands to indicate that the copied or moved line(s) appear before the line marked with “B”.

See also “A (after)” on page 39 and “O (overlay)” on page 49.

B

Example

```
b0110  .WHEN MAP="PF2"
m0120 ..INSERT RECORD
```

BIND

Primary command that converts a program from unbound to bound (or bound to unbound) format.

```
     ⎡ON ⎤
BIND ⎢   ⎥ [n]
     ⎣OFF⎦
```

Example

```
BIND 2
```

BOTTOM

Primary command that moves your terminal window to the bottom of your program.

See also “TOP” on page 55, “DOWN” on page 43, “UP” on page 55, “RIGHT” on page 52, and “LEFT” on page 46.

```
⎧BOTTOM⎫
⎩BOT   ⎭
```

Example

```
BOT
```

C (copy)

Line command that copies one line or multiple lines within your program.

See also “COPY” on page 42, “M (move)” on page 48, “A (after)” on page 39, “B (before)” on page 39, and “O (overlay)” on page 49.

```
⎧C ⎫
⎨Cn⎬
⎩CC⎭
```

Example

```
00001  ENTRY COMPOUND_INTEREST
cc 02  .SHOW "WHAT IS THE CAPITAL AMOUNT?"
cc 03  .OBTAIN SAVINGS
a 004  .IF SAVINGS<=0
00005  .END
00006  EXIT
```

CANCEL

Primary command that terminates editing mode without saving the program you are editing. MANTIS returns you to the Program Design Facility menu, the EDIT parameter entry screen, or the EEPR program directory list, depending on which you used to enter FSE. If issued at *n*-level edit, MANTIS returns you to previous edit level.

See also “END” on page 43, “KILL” on page 46, and “QUIT” on page 50.

$$\left\{\begin{array}{l}\textbf{CAN[CEL]}\\ \textbf{PA2}\end{array}\right\}$$

Example

```
CANCEL
```

CHANGE

Primary command that alters a text string or text strings in a program. (To repeat a CHANGE command, use RCHANGE.)

```
{ CHANGE }        [ " ] { fromstring } [ " ]         [ " ] { tostring } [ " ]
{ CHG    }  [,]   [   ] {            } [   ]  [,]    [   ] {          } [   ]  [,]
{ C      }        [ , ] {     *      } [ ' ]         [ ' ] {    *     } [ ' ]

[ NEXT  ]
[ PREV  ]         [ CHARS    ]
[ FIRST ]  [,]    [ PRE[FIX] ]
[ LAST  ]         [ SUF[FIX] ]
[ ALL   ]         [ WORD     ]
```

Example

```
c "Index" LIST
```

COPY

Primary command that copies portions of another program into the program in the work area. You can copy lines from the program currently in the work area or in a library.

See also “C (copy)” on page 40.

```
                                                   [ FIRST ]                [ FIRST     ]
COPY[[user - name :]program - name[/password ],]   [ n1    ] [ ,n2   ]  b   [ AFTER n3  ]
                                                   [ LAST  ] [ ,LAST ]      [ LAST      ]
```

Example

```
COPY SIMPLE_INTEREST,70,80 AFTER 34
```

D (delete)

Line command that specifies one or more lines for deletion.

See also “ERASE” on page 44 and “PURGE” on page 31.

$$\left\{\begin{array}{l}\textbf{D}\\ \textbf{D}\boldsymbol{n}\\ \textbf{DD}\end{array}\right\}$$

Example

```
d 05 .COUNTER=1
```

DOWN

Primary command that scrolls toward the end of a program listing.

See also “BOTTOM” on page 40, “LEFT” on page 29, “RIGHT” on page 32, “UP” on page 55, and “TOP” on page 55.

DOWN [*n*]

Example

```
DOWN 15
```

END

Primary command that saves or replaces your modified program and exits from programming mode, or returns to the previous edit level for *n*-level editing.

See also “CANCEL” on page 22, “KILL” on page 46, “LOGOFF” on page 30, “MENU” on page 30, and “QUIT” on page 50.

END

Example

```
END
```

ERASE

Primary command that deletes one or more program lines where *n*2 is the ending line number and *n*3 is the number of lines.

See also “D (delete)” on page 43 and “PURGE” on page 31.

$$\textbf{ERASE}\ \boldsymbol{n1}\begin{bmatrix} ,\boldsymbol{n2} \\ ;\boldsymbol{n3} \end{bmatrix}$$

Example

```
ERASE 50,70
```

ERRCODE

Primary command that displays text for the three-character syntax error messages you receive in FSE.

See also “HELP” on page 28.

$$\left\{\begin{matrix} \textbf{ERRCODE} \\ \textbf{EC} \end{matrix}\right\}\boldsymbol{xxx}$$

Example

```
ERRCODE NCQ
```

FIND

Primary command that locates and displays an occurrence of a text string in a program.

See also “LOCATE” on page 47.

$$\left\{\begin{matrix}\textbf{FIND}\\ \textbf{F}\end{matrix}\right\}\ \textbf{[,]}\ \begin{bmatrix}\textbf{"}\\ \textbf{'}\end{bmatrix}\ \left\{\begin{matrix}\textbf{\textit{string}}\\ \boldsymbol{*}\end{matrix}\right\}\ \begin{bmatrix}\textbf{"}\\ \textbf{'}\end{bmatrix}\ \textbf{[,]}\ \begin{bmatrix}\underline{\textbf{NEXT}}\\ \textbf{PREV}\\ \textbf{FIRST}\\ \textbf{LAST}\\ \textbf{ALL}\end{bmatrix}\ \textbf{[,]}\ \begin{bmatrix}\underline{\textbf{CHARS}}\\ \textbf{PRE[FIX]}\\ \textbf{SUF[FIX]}\\ \textbf{WORD}\end{bmatrix}$$

Example

```
F "FIRST" WORD
```

HELP

Primary command that provides an explanation of an error message, a command, a list of reserved words or on-line help for FSE. For statements used with END (WHILE-END, IF-END), do not specify “END” in conjunction with the HELP command (use HELP WHILE or HELP IF).

$$\textbf{HELP}\ \left\{\begin{matrix}\textbf{RESERVED}\\ \textbf{\textit{command - name}}\\ \textbf{CODE\_}\ \textbf{\textit{xxx}}\\ \textbf{FSE}\\ \textbf{HELP}\end{matrix}\right\}$$

Example

```
HELP CONVERSE
```

I (insert)

Line command that inserts blank lines in a program.

I [*n*]

Example

```
I5   04   .GET RECORD FIRST
```

KILL

Primary command that terminates a program currently paused (waiting for data), or a program in a loop. Whenever a program executes a WAIT, OBTAIN or CONVERSE statement, or it has just issued the message “POTENTIAL PROGRAM LOOP ENCOUNTERED”, you can stop program execution by issuing KILL.

See also “CANCEL” on page 22, “END” on page 43, and “QUIT” on page 50.

KILL

Example

```
KILL
```

LEFT

Primary command that scrolls your terminal the specified amount toward column one in a program listing.

See also “RIGHT” on page 32, “TOP” on page 55, “BOTTOM” on page 40, “DOWN” on page 43, and “UP” on page 55.

LEFT *n*

Example

```
LEFT 15
```

LIST

Primary command that lists all or part of the program currently in the work area. MANTIS removes insignificant blanks and indents nested conditions (with periods) according to hierarchy.

LIST [*n1*]

Example

```
LIST
```

LOAD

Primary command that retrieves an existing program from a library and places it in your current work area.

LOAD ["] [*user - name*:] [*program - name*] [/*password*] ["]

Example

```
LOAD EXAMPLE_PROGRAM
```

LOCATE

Primary command that finds a specific line in the current program and lists the program beginning at this line.

See also “FIND” on page 45.

$$\left\{\begin{array}{l}\textbf{LOCATE}\\ \textbf{LOC}\\ \textbf{L}\end{array}\right\}\ \boldsymbol{n}$$

Example

```
LOC 170
```

LOGOFF

Primary command that saves or replaces a program or component (if changes have been made) and exits from MANTIS, or returns to the previous edit level for *n*-level editing.

LOGOFF

Example

```
LOGOFF
```

M (move)

Line command that moves a line or lines within a program.

See also "A (after)" on page 39, "B (before)" on page 39, "C (copy)" on page 40, and "O (overlay)" on page 49.

$$\left\{\begin{array}{l}\textbf{M}\\ \textbf{M}\boldsymbol{n}\\ \textbf{MM}\end{array}\right\}$$

Example

```
a 040   .WHILE RECORD<>"END"
00050   .END
m 060   .CONVERSE MAP
```

MENU

Primary command that saves or replaces a program or component (if changes have been made) and displays your MANTIS Facility Selection menu, or returns to the previous edit level for *n*-level editing.

MENU

Example

```
MENU
```

NEW

Primary command that clears the current work area.

NEW

Example

```
NEW
```

O (overlay)

Line command used with the COPY or MOVE line commands to specify the destination of the program lines. Destination line(s) are erased and replaced by the copied or moved line(s).

See also “B (before)” on page 39 and “A (after)” on page 39.

$$\left\{\begin{array}{l}\textbf{O}\\ \textbf{O}\boldsymbol{n}\\ \textbf{OO}\end{array}\right\}$$

Example

```
cc040 .WHILE RECORD<>"END"
00050 ..DO SUBR(A)
cc060 .END
.
.
.
o0170 .DO SUBR(A)
```

PRINT

Primary command that routes the program in your program work area to the printer designated in your User Profile.

PRINT

Example

```
PRINT
```

PROFILE

Primary command that displays and permits temporary changes to the edit profile for the duration of the current session.

$$\left\{\begin{array}{l}\textbf{PROFILE}\\ \textbf{PROF}\end{array}\right\}$$

Example

```
PROF
```

PURGE

Primary command that erases a program from a library (but not from the current work area).

See also “D (delete)” on page 43 and “ERASE” on page 44.

PURGE ["] [*program - name*] [/*password*] ["]

Example

```
PURGE EXAMPLE_PROGRAM
```

QUIT

Primary command that terminates programming mode and returns you to the previous panel.

See also “CANCEL” on page 22, “END” on page 43, and “KILL” on page 46.

QUIT

Example

```
QUIT
```

R (repeat)

Line command that copies a line or lines in a program immediately following the line(s) you mark.

See also “C (copy)” on page 40 and “M (move)” on page 48.

$$\left\{\begin{array}{l}\textbf{R}\\ \textbf{R}\boldsymbol{n}\\ \textbf{RR}\\ \textbf{RR}\boldsymbol{n}\end{array}\right\}$$

Example

```
0r009  EXIT
```

RCHANGE

Primary command that repeats the last CHANGE command you issued.

$$\left\{\begin{array}{l}\textbf{RCHANGE}\\ \textbf{RCHG}\\ \textbf{RC}\end{array}\right\}$$

Example

```
RC
```

REPLACE

Primary command that replaces a program in your library with the program currently in your work area.

See also “SAVE” on page 53.

REPLACE ["][*program - name*] [*/password*] [*/description*] ["]

Example

```
REPLACE DATAENTRY
```

RESET

Primary command that resets any pending, ambiguous, or invalid primary commands, line commands and commands issued from PF keys.

$$\left\{ \begin{array}{l} \textbf{RESET} \\ \textbf{RES} \end{array} \right\}$$

Example

```
RESET
```

RFIND

Primary command that repeats the last FIND command you issued. Since RFIND searches from the current cursor position, the default PF keys, PF5 and PF17, for RFIND are provided.

$$\left\{ \begin{array}{l} \textbf{RFIND} \\ \textbf{RF} \end{array} \right\}$$

Example

```
RF
```

RIGHT

Primary command that scrolls your display toward the highest column on your terminal in a program listing.

See also “LEFT” on page 29, “TOP” on page 55, “BOTTOM” on page 40, “DOWN” on page 43, and “UP” on page 55.

RIGHT *n*

Example

```
RIGHT 15
```

RUN

Primary command that executes the program currently in the work area. If a value is supplied for *n*, MANTIS begins running the program at that number.

RUN [*n*]

Example

```
RUN
```

S (select)

Line command that selects a program or component named in the COMPONENT, REPLACE, or SOURCE statements for editing in FSE.

S

Example

```
s0110 .COMPONENT"ACCT:CUST_INIT/PASSWORD"
```

SAVE

Primary command that saves the program currently in the work area into a library.

See also "REPLACE" on page 51.

SAVE ["] [*program - name*] [/*password*][/*description*] ["]

Example

SAVE DATAENTRY

SCROLL

Primary command that determines the scrolling mode of the terminal, or scrolling increments for window mode.

SCROLL [**OFF**
ON
[*row*] [,*col*]]

Example

```
SCROLL OFF
```

SEQUENCE

Primary command that renumbers the program lines currently in the work area. MANTIS assigns *n*1 to the first statement and increments by *n*2 for each succeeding statement.

SEQUENCE [*n1*[, *n2*]]

Example

```
SEQUENCE 100
```

TOP

Primary command that moves your terminal window to the top of your current program.

See also “UP” on page 55, “DOWN” on page 43, “LOCATE” on page 47, “BOTTOM” on page 40, “LEFT” on page 29, and “RIGHT” on page 32.

TOP

Example

```
TOP
```

UP

Primary command that scrolls your display the specified amount toward line one in a program listing.

See also “TOP” on page 55, “LOCATE” on page 47, “DOWN” on page 43, “BOTTOM” on page 40, “LEFT” on page 29, and “RIGHT” on page 32.

UP [*n*]

Example

```
UP 15
```

3

MANTIS statements, functions, and commands

ABS Function

Returns the absolute value of an arithmetic expression.

ABS(*a*)

Example

```
ABS(0)       returns  0
ABS(-13E9)   returns .13E11
```

ACCESS Statement

Identifies an external file to be accessed by your program. MANTIS retrieves the file view description from the user library and places it in the work area. If view variables are not already defined, MANTIS defines them. If view variables are already defined, MANTIS checks for consistency.

ACCESS *name1*([*library1*:]*access - name1*, *password1*[, PREFIX][, *n1*])
[, *name2*([*library2*:]*access - name2*, *password2*[, PREFIX][, *n2*])...]

Example

```
ACCESS RECORD("INDEX","IDXPSWD",16)
```

ASI (Attribute Status Indicator) Function

Returns the status of a field in an RDM logical record.

ASI(*view - name*,*field - name*)

Example

```
ASI(PARTS,PART_NAME)
```

ATN Function

Returns the angle in radians whose tangent is the arithmetic expression (*a*).

ATN(*a*)

Example

```
ATN(10)    returns   1.47112767430373
ATN(100)   returns   1.56079666010823
```

ATTRIBUTE Statement

Changes the attributes of a panel, a field on a panel, a terminal, or a printer. The changed attributes remain in effect until you change them again or use the RESET attribute to revert to the original (default) specification.

$$\textbf{ATTRIBUTE}\left\{\begin{array}{ll}\textbf{(}\textbf{\textit{screen - name}},\left[\begin{array}{l}\textbf{,}\textbf{\textit{field - name}}\\ \textbf{,(}\textbf{\textit{row}},\textbf{\textit{col}}\textbf{)}\end{array}\right]\textbf{)} & \\ \textbf{(PRINTER)} & = \textbf{\textit{e1}}[,\textbf{\textit{e2}},\textbf{\textit{e3}},\ldots] \\ \textbf{(TERMINAL)} & \\ \textbf{(TERMINA, CURSOR) ="(}\textbf{\textit{row}},\textbf{\textit{col}}\textbf{)"} & \end{array}\right\}$$

Example

```
ATTRIBUTE(INVOICE,ACCT_NUM)="BRIGHT,PROTECTED"
```

(See “Attribute types” on page 127 for a list of attributes used by the ATTRIBUTE statement.)

TERMINAL and CURSOR versions may be restricted by your installation.

ATTRIBUTE Function

Returns attributes of a field or device or the physical coordinates of the cursor on the panel. Your Master User determines access to the ATTRIBUTE function.

$$\textbf{ATTRIBUTE} \left\{ \begin{array}{l} \mathbf{(}\textbf{\textit{screen - name}}\mathbf{,}\begin{bmatrix} \textbf{\textit{field - name}} \\ \mathbf{(}\textbf{\textit{row}}\mathbf{,}\textbf{\textit{col}}\mathbf{)} \end{bmatrix}\mathbf{)} \\ \textbf{(PRINTER)} \\ \textbf{(TERMINAL)} \\ \textbf{(TERMINAL, CURSOR)} \end{array} \right\}$$

Example

```
SHOW ATTRIBUTE(MAP,TEST_FIELD) returns
  "(9,2),14,TXT,UPP,UNP,REV,RED,AUT"
```

(See “ATTRIBUTE function (values returned)” on page 131 for a list of attributes returned by the ATTRIBUTE function.)

BIG Statement

Names and supplies the dimensions for numeric variables. MANTIS creates an eight-byte numeric field (or an array of 8-byte fields) and associates it with the name you specify.

BIG *name1*[(*n1*[, *n2*])][, *name2*[(*n1*[, *n2*])]]. . .

Example

```
BIG ALPHA(64,3),BETA(12)
```

BREAK Statement

Use the BREAK statement to exit from a FOR-END, UNTIL-END, WHEN-END, or WHILE-END statement. The statement after the END statement is executed next.

BREAK

Example

The following example shows how the BREAK statement can be used to exit a FOR-END condition:

```
10 FOR L=1 TO MAXLINES:|   For each screen line
20 .GET CUSTOMER LEVEL=L:|Get customer detail fields
30 .IF CUSTOMER="END":|Check status from GET
40 ..BREAK:|Exit FOR Loop if end of file
50 .END  to statement 70.
60 END
70 CONVERSE CUST_DETAILS:|Display customer details
```

CALL Statement

Invokes an interface program. MANTIS calls the program specified in the interface profile and sets the symbolic name variable equal to the status returned by the program.

CALL *interface*[(*e1*,*e2*. . .)][LEVEL = *n*]

Example

```
CALL MASTER("GET",1234) LEVEL=2
```

CHAIN Statement

Replaces the program currently executing with another MANTIS program and begins executing that program. MANTIS terminates the issuing program and erases all variables, except those being passed.

CHAIN" [*library*:] *program - name*" [, *name1*, *name2*, . . .] [LEVEL]

Example

```
CHAIN "GAMES_MENU"
```

CHR Function

Use the CHR (character) function to return a text value consisting of the character corresponding to the EBCDIC code specified.

CHR(*a*)

Example

```
CHR(97) returns /
CHR(129) returns a
```

CLEAR Statement

Use the CLEAR statement to clear the scroll map display, clear the data referred to by the symbolic name of a complex entity, or a specific variable name, or clear all program data.

$$\textbf{CLEAR}\begin{bmatrix} \textbf{\textit{name}}\ \ \textbf{[,}\ldots\textbf{]} \\ \textbf{ALL} \end{bmatrix}$$

Example

```
CLEAR MAP
```

COMMIT Statement

Indicates the completion of a Logical Unit of Work (LUW), or toggle automatic COMMIT processing with the COMMIT statement. COMMIT with no parameters commits pending database updates and prevents them from being backed out by RESET or a system failure. COMMIT with no parameter also flushes updated buffers for the MANTIS cluster and the external files.

$$\textbf{COMMIT}\begin{bmatrix}\textbf{ON}\\ \textbf{OFF}\end{bmatrix}$$

Example

```
COMMIT
```

COMPONENT Statement

When coded in a MANTIS source program, the COMPONENT statement identifies each component that can be assembled by the Compose action into expanded component code in a composed (executable) program. When displayed in a MANTIS composed executable program, the COMPONENT statement identifies each component that can be nominated and disassembled by the Decompose action into separate, updated components.

NOTE

If UPPERCASE = N has been specified in FSE, you must enter this statement in UPPERCASE mode for it to be recognized by MANTIS.

COMPONENT"[*library*:] *component - name* [/*password*] [/*description*]"

Example

```
COMPONENT"ACCT:CUST_ERROR_PROC"
```

CONVERSE Statement

Sends a formatted panel design or map set to a terminal and returns any response or alterations to the program.

$$\textbf{CONVERSE } \textbf{\textit{screen - name}} \begin{bmatrix} \\ \textbf{[(}\textbf{\textit{row1, col1}}\textbf{)]} \\ \textbf{RELEASE} \end{bmatrix} \begin{bmatrix} \textbf{WAIT} \\ \textbf{SET} \\ \textbf{UPDATE} \end{bmatrix}$$

Example

```
CONVERSE MAP2(5,10)WAIT
```

COS Function

Returns the cosine of *a* where *a* is in radians.

COS(*a*)

Example

```
Y=COS(X)
```

CSIOPTNS Statement

Specifies the values of three options used to execute the Compose action.

NOTE

If UPPERCASE = N has been specified in FSE, you must enter this statement in UPPERCASE mode for it to be recognized by MANTIS.

$$\textbf{|*CSIOPTNS" [COMMENTS =} \begin{bmatrix} \underline{\textbf{YES}} \\ \textbf{NO} \end{bmatrix} \textbf{] [:FORCE =} \begin{bmatrix} \textbf{YES} \\ \underline{\textbf{NO}} \end{bmatrix} \textbf{] [:SEQUENCE [}\textbf{\textit{n, n}}\textbf{]]"}$$

Example

```
CSIOPTNS"COMMENTS=NO:FORCE=YES:SEQUENCE 5,5"
```

CURSOR Function

Indicates whether the cursor appeared in a specific field at the last terminal I/O. MANTIS performs the test and returns one of the following:

- TRUE
- FALSE
- The field symbolic variable name of the entity that contained the cursor at the last CONVERSE
- The screen symbolic variable name of the entity that contained the cursor at the last CONVERSE

$$\textbf{CURSOR}\left\{\begin{array}{l} \textbf{(}\boldsymbol{\mathit{screen\ \text{-}\ name}}\textbf{,}\left\{\begin{array}{l}\boldsymbol{\mathit{field\ \text{-}\ name}}\\ \textbf{(}\boldsymbol{\mathit{row,col}}\textbf{)}\end{array}\right\}\\ \textbf{("FIELD")}\\ \textbf{("SCREEN")}\\ \\ \end{array}\right\}$$

Example

```
CURSOR(MAP,CUST_NO)
```

DATAFREE Function

Returns the number of bytes remaining in your data area. The data area is used to hold the values for all the variables in your program. The amount available decreases as additional variables are defined.

DATAFREE

Example

```
SHOW DATAFREE
```

DATE Statement

Sets up the format of a text string that the DATE function uses to return the current date.

DATE = *mask - expression*

Example

```
DATE="YYYY/MM/DD"
```

DATE Function

Returns a text string containing the current date. The format of the date is determined by a previous DATE= mask or the system default.

DATE

Example

```
SHOW DATE
1998/01/01
```

DBCS Statement (Kanji users only)

The DBCS statement names and specifies dimensions for DBCS variables and lists.

NOTE

This function is only available for release 5.2 service level 5231 and above.

DBCS *name1*[(*n1*[,*n2*])]

[,*name2*[(*n1*[,*n2*])] . . .]

Example

NOTE

In this example, < indicates SO or Shift Out, > indicates SI or Shift In.

```
00010 DBCS FIELDK(5),ARRAYK(3,20)
00020 FIELDK=K" %% ":ARRAYK(1)=K" %%%% ":ARRAYK(2)=G "<%%%%>"
00030 SCREEN MAP("DBCS_MAP","PSW")
00040 WHILE MAP<>"CANCEL"
00050
00060
00070
    .
    .
    .
```

DELETE Statement

Deletes a record from a MANTIS file, an external file, a personal computer file, an RDM logical view, or a TOTAL file. Before you delete a record from a file or view, you must first open it by processing the associated FILE, TOTAL, ACCESS, or VIEW statement. You do not need to GET a MANTIS or external file record before deleting it. You must read an RDM logical view before deleting it. MANTIS returns a text string, in either the *file-name* or *view-name* variable, that reflects the operation's status.

MANTIS file, External file

$$\textbf{DELETE}\ \textbf{\textit{file - name}}\begin{bmatrix}\textbf{(\textit{key1}, \textit{key2}, ...) ALL}\\ \textbf{LEVEL = \textit{n}}\\ \textbf{ALL}\end{bmatrix}$$

Personal Computer file, Total file

DELETE *file - name*[LEVEL = *n*]

RDM Logical view

DELETE *view-name*[ALL][LEVEL=*n*]

Example

```
DELETE RECORD LEVEL=COUNTER
```

DEQUEUE Statement

Releases control of a resource (e.g., a program, file, and so on), or, for VSAM external file users, releases a previously reserved external file record, or for TOTAL users, releases a previously reserved TOTAL database record. MANTIS also releases any program or TOTAL record that is waiting while other tasks use a resource.

See also "GET Statement" on page 74 and "ENQUEUE Statement " on page 69.

$$\textbf{DEQUEUE}\begin{Bmatrix} \textbf{\textit{resource}} \\ \textbf{\textit{file - name}} \end{Bmatrix}$$

Example

```
DEQUEUE "CUSTOMERS"+RECORD_KEY
```

DO Statement

Transfers program execution to an internal or external subroutine. A subroutine is a block of statements either within the existing MANTIS program, or identified by a PROGRAM statement, that performs a function required at one or more points in a program. After executing the subroutine and upon encountering an EXIT statement, execution returns to the next program line following the DO statement.

DO *entry - name*[(*argument1*,*argument2*,...)]

Example

```
DO ERROR_RTN(CUST_NO)
```

DOLEVEL Function

Returns the current execution level in an external subroutine.

DOLEVEL

Example

```
SHOW DOLEVEL
  2
```

E Function

Returns the value of natural E (2.71828182).

E

Example

```
00010 X=E
```

The example above sets the variable *x* to 2.71828182.

ENQUEUE Statement

Holds control of a resource as identified by the resource-string. Any subsequent ENQUEUE on that resource by another program causes that program to remain in a wait state until the resource is released.

See also “GET Statement” on page 74 and “DEQUEUE Statement” on page 68.

ENQUEUE *text or DBCS-expression*

Example

```
ENQUEUE "CUSTOMERS"+RECORD_KEY
```

ENTRY - EXIT Statement

Defines the boundary of a subroutine or top level routine of a program. When a DO or CHAIN statement invokes a subroutine or program bound by an ENTRY-EXIT, the arguments (and all references to them) passed by the DO or CHAIN statement replace the subroutine's arguments. EXIT is also a command to the editor.

ENTRY ***entry - name*** **[(*parameter1*, *parameter2*, . . .)]**
.
. ***statements***
.
EXIT

Example

```
ENTRY INSERT_RECORD
.INSERT REC
EXIT
```

EXEC_SQL-END Statement

Allows SQL statements to be executed in a MANTIS program.

EXEC_ SQL **or** **EXEC_ SQL(*nn*)**

Example

```
BIG EMP_NUM
TEXT EMP_NAME(30)
EXEC_SQL
. | SELECT EMPNO, EMPNAME
. | INTO : EMPL_NUM, EMPL_NAME
. | WHERE EMPNO = : EMPL_NUM
END
```

EXIT Command/Statement

Returns control from an external routine to the invoking program. The EXIT statement also bounds the subroutine started by the corresponding ENTRY statement.

EXIT

Example

```
EXIT
```

EXP Function

Returns the value of natural E to the power of a valid arithmetic expression.

EXP(*a*)

Example

```
EXP(100)
```

This example returns a result of .268811714181613E44.

FALSE Function

MANTIS constant that returns the value zero.

FALSE

Example

```
ERROR=FALSE
```

FILE Statement

Identifies a MANTIS internal file that your program accesses. MANTIS retrieves the file description from your library and places it in your work area. If file variables are not already defined, MANTIS defines them. If file variables are already defined, MANTIS checks for consistency.

FILE *name1*([*library1*:]*file - name1*, *password1* [, PREFIX][, *n1*])
[, *name2*([*library2*:]*file - name2*, *password2*[, PREFIX][, n*2*])...]

Example

```
FILE RECORD("INDEX","SERENDIPITY",16)
```

FOR-END Statement

Execute a block of statements repeatedly while a counter is incremented or decremented through a specified range of values.

FOR *counter=initial* TO *final* [BY *increment*]

```
       .
       statements
       .
 END
```

Example

The following example shows the FOR-END statement using literals. Note that the statement within the loop is executed five times, and the loop counter (I) will be equal to 6 when statement 50 is executed.

```
10 FOR I = 1 TO 5 BY 1
20 .SHOW I
30 . (statements)
40 END
50 WAIT
```

FORMAT Function

Returns a text string conversion of a numeric expression, or a formatted output for SHOW, according to the supplied edit mask. This function also allows you to test panel design masks for expected results.

FORMAT(*numeric - expression*, *mask*, [*digit - select - character*])

Example

```
SHOW FORMAT(100,"$$#Z.##")  returns  $100.00
```

FSI Function

Indicates the success or failure of a logical view, MANTIS file, or external file GET, DELETE, INSERT, RELEASE, or UPDATE.

FSI(*name*[, *msg*])

Example

```
IF FSI(CUSTOMERS)<>"GOOD"
```

GET Statement

The GET statement reads a record from a MANTIS file, an external file, a personal computer file, an RDM logical view or a TOTAL file. Before you can read from a file or view, you must open it by processing the associated FILE, ACCESS, VIEW, or TOTAL statement. MANTIS returns a text string, in either the *file-name* or *view-name* variable, that reflects the operation's status.

MANTIS file, External file, PC file

```
              ┌ (key1,key2,;;;)[EQUAL] ┐
              │       FIRST            │
GET file-name │       NEXT             │ [ENQUEUE]** [LEVEL=n]
              │       PRIOR *          │
              └       LAST *           ┘
```

* Not supported in the IMS environment.
** Not supported in PC environment.

RDM Logical view

```
                                     ┌ NEXT  ┐
               ┌ (key1,key2,...)   ┐ │ PRIOR │
GET view - name│ AT mark - name    │ │ FIRST │ [ENQUEUE][LEVEL = n]
               └ SAME              ┘ └ LAST  ┘
```

TOTAL file

```
                ┌                    ┌ AT refer     ┐                       ┐
                │                    │ BEFORE refer │                       │
                │ SET (key1,key2,...)│ AFTER refer  │[ENQUEUE] [LEVEL = n]  │
GET file - name │                    │ FIRST        │                       │
                │                    └ LAST         ┘                       │
                │┌ (key1,key2,...) ┐                                        │
                └└ [FIRST]         ┘                                        ┘
```

Example

```
GET RECORD("WILLIAMS") LEVEL=BUFFER
```

HEAD Statement

Centers a heading on the top line of an unformatted screen and sets it to high intensity.

HEAD *heading*

Example

```
HEAD "BUZZ PHRASE GENERATOR"
```

HELP Command

Provides further explanation of an error message, a command, or a list of reserved words. For statements used with END (e.g., WHILE-END, IF-END), do not specify “END” in conjunction with the HELP command (use HELP WHILE or HELP IF).

$$\textbf{HELP}\left\{\begin{array}{l}\textbf{RESERVED}\\ \textbf{\textit{command - name}}\\ \textbf{CODE\_ \textit{xxx}}\\ \textbf{FSE}\\ \textbf{HELP}\end{array}\right\}$$

Example

```
HELP CONVERSE
```

IF-ELSE-END Statement

Executes a block of statements only if a specified condition (or conditions) is true.

```
 IF expression
     .blocka
[ELSE
    .blockb]
 END
```

Example

```
..IF REC="FOUND"
...UPDATE REC
..ELSE
...MSG="CUSTOMER NOT FOUND"
..END
```

INSERT Statement

Inserts a new record into a MANTIS file, an external file, a personal computer file, an RDM logical view, or a TOTAL file. If you don't read a record before inserting, then RDM logical view performs the insert relative to the current record position. MANTIS returns a text string, in either the *file-name* or *view-name* variable, that reflects the operation's status.

MANTIS file, External file, PC file

INSERT *file - name*[LEVEL = *n*]

RDM Logical view

$$\textbf{INSERT } \textbf{\textit{view - name}}\begin{bmatrix} \underline{\textbf{NEXT}} \\ \textbf{PRIOR} \\ \textbf{FIRST} \\ \textbf{LAST} \end{bmatrix}[\textbf{LEVEL} = \textbf{\textit{n}}]$$

TOTAL file

$$\textbf{INSERT } \textbf{\textit{file - name}}\begin{bmatrix} \textbf{FIRST} \\ \underline{\textbf{LAST}} \\ \textbf{BEFORE refer} \\ \textbf{AFTER refer} \end{bmatrix}[\textbf{LEVEL} = \textbf{\textit{n}}]$$

Example

```
INSERT RECORD LEVEL=COUNTER
```

INT Function

Returns the integer value of *a* where *a* is any arithmetic expression.

INT(*a*)

Example

```
INT(45.5) returns 45
```

INTERFACE Statement

Specifies an interface that your program accesses. MANTIS retrieves the interface description from your library and places it in your work area. If interface variables are not already defined, MANTIS defines them. If interface variables are already defined, MANTIS checks for consistency.

INTERFACE *name1*([*library1*:]*file - name1*, *password1* [, PREFIX][, *n1*])
[, *name2*([*library2*:]*file - name2*, *password2*[, PREFIX][, *n2*])...]

Example

```
INTERFACE MASTER("CUSTOMERS","ALIBABA",10)
```

KANJI Statement (Kanji users only)

Names and specifies dimensions for Kanji variables and lists.

KANJI *name1*[(*n1*[, *n2*])][, *name2*[(*n1*[, *n2*])...]

Example

```
KANJI FIELDK(5),ARRAYK(3,20)
```

KEY Function

Returns a text string that identifies the key you pressed in response to a CONVERSE, OBTAIN, PROMPT or WAIT statement.

KEY

Example

```
WHILE KEY<>"CANCEL"
```

KILL Command

Terminates a program listing (line editor), a program currently paused (waiting for data), or a program in a loop. Whenever a program executes a WAIT, OBTAIN, or CONVERSE statement, or has just issued the message “”POTENTIAL PROGRAM LOOP ENCOUNTERED”, you can stop program execution by entering KILL.

KILL

Example

```
KILL
```

LANGUAGE Statement

Changes the current language code for the signed on user. (Not supported in all environments.) The language code determines which messages, facility screens, and help prompters are displayed on the terminal.

LANGUAGE = e

Example

```
LANGUAGE="ITA"
```

LANGUAGE Function

Returns the current language code for the signed on user. (Not supported in all environments.)

LANGUAGE

Example

```
IF  LANGUAGE="ITA"
      .
      .
      .
END
```

LET Statement (Numeric variables)

Assigns a value to a variable (or variables) or any array (or arrays). MANTIS evaluates the expression and sets the variable or array equal to that value.

$$\textbf{[LET]}\,\boldsymbol{v}\begin{bmatrix}(\boldsymbol{i})\\(\boldsymbol{i},\boldsymbol{j})\end{bmatrix}\quad\textbf{[ROUNDED(}\boldsymbol{n}\textbf{) = }\boldsymbol{e1}\ \textbf{[},\boldsymbol{e2},\boldsymbol{e3}\ldots\textbf{]}$$

Example

```
LET ANSWER ROUNDED(2)=CAPITAL*(1+RATE/100)**LENGTH
```

LET Statement (Text/Kanji variables)

$$\textbf{[LET]}\,\boldsymbol{v}\begin{bmatrix}(\boldsymbol{x})\\(\boldsymbol{x},\boldsymbol{y})\\(\boldsymbol{i},\boldsymbol{x},\boldsymbol{y})\end{bmatrix}\quad = \ \boldsymbol{e1}\ \textbf{[},\boldsymbol{e2},\boldsymbol{e3}\ldots\textbf{]}$$

Example

```
LET VAR1(1,12)="GOOD MORNING"
```

LOG Function

Returns the natural logarithm of a positive arithmetic expression.

LOG(*a*)

Example

```
SHOW LOG(X):WAIT
```

LOWERCASE Function

Converts a text string into lowercase.

LOWERCASE(*t*)

Example

```
SHOW LOWERCASE("abc $ ABC"):WAIT

abc $ abc
```

LUID Function

Returns an 8-character text string containing the VTAM logical unit ID (netname).

LUID

Example

```
IF LUID="NMMAI032"
…
```

MARK Statement (SUPRA RDM users only)

Obtains the current position of the logical view as established by the last GET, UPDATE, or INSERT statements. Before you can mark a view, you must open the file by processing the associated VIEW statement.

MARK *view - name* AT *mark - name* [LEVEL = *n*]

Example

```
MARK CUSTOMER AT CUST_MARK
```

MIXD Function

Extracts Kanji DBCS (Double Byte Character Set) data from mixed data.

MIXD(*t*)

Example

```
MIXMODE ON
TEXT ALPHA(20)
KANJI GAMMA(20)
ALPHA="A<%1>BC<%2>"
GAMMA=MIXD(ALPHA)

GAMMA returns %1%2
```

MIXM Function

Converts a Kanji or DBCS expression to a mixed data text string containing shift codes from Kanji or DBCS data.

MIXM(*name*)

Example

```
MIXMODE ON
TEXT ALPHA(20)
KANJI GAMMA(20)
GAMMA=K" %1%2 "
ALPHA=MIXM(GAMMA)

ALPHA contains <%1%2>
```

MIXMODE Statement

Controls the handling of mixed data. MIXMODE ON sets the program in mixed-data mode. MIXMODE OFF sets the program in nonmixed-data mode.

MIXMODE $\begin{bmatrix}\textbf{ON}\\ \underline{\textbf{OFF}}\end{bmatrix}$

Example

```
MIXMODE ON
TEXT A(80),B(80)
A="abc<%1%2>de<%3>fg"
B=A-"<%2>":|B contains   abc<%1>de<%3>fg
MIXMODE OFF
B=A-"<%2>":|B contains   abc<%1%2>de<%3>fg
```

MIXT Function

Extracts an SBCS (Single Byte Character Set) text string from text and mixed-data expressions.

MIXT(*t*)

Example

```
MIXMODE ON
TEXT ALPHA(20),BETA(20)
ALPHA="A<%1>BC<%2>"
BETA=MIXT(ALPHA)

BETA contains ABC
```

MODIFIED Function

Tests whether a specific field, any field within a map definition, or any field within the entire map set changed during the last physical I/O. Because zero evaluates to FALSE, you can use MODIFIED as a logical or arithmetic function.

$$\textbf{MODIFIED}\left\{\begin{array}{l}(\textbf{\textit{screen - name}}\ [,\textbf{\textit{field - name}}])\\(\textbf{TERMINAL})\end{array}\right\}$$

Example

```
IF MODIFIED(CLIENT_INFO)
```

NEXT Statement

Proceeds immediately to the next conditional repeat in a FOR-END, UNTIL-END, or WHILE-END statement or to the next WHEN condition in a WHEN-END statement.

NEXT

Example

```
  10 FOR L=1 TO MAXLINES
  20 .IF NOT(MODIFIED(CUST_DETAIL))   <----If this condition is TRUE, then
  30 . NEXT                                 logic flow will continue to
  40 .END                                   statement 10.
  50 .UPDATE CUSTOMER_FILE
  60 END
```

NOT Function

Returns TRUE (1) for an arithmetic expression if *a* evaluates to FALSE (0); otherwise, NOT returns FALSE (0).

NOT(*a*)

Example

```
IF NOT (A=3 OR J=1)
```

NULL Function

Returns a null (zero-length) text, DBCS, or Kanji value (“” or K“”).

NULL

Example

```
TEXT A
A=NULL
IF A=NULL
.
.
.
END
```

NUMERIC Function

Use the NUMERIC function to determine if a text expression contains a valid number.

NUMERIC (*text-expression*)

Example

```
NUMERIC("123,456.789")   returns  TRUE
NUMERIC("$1234.55")      returns  FALSE
NUMERIC("-.05")          returns  TRUE
NUMERIC ("")             returns  FALSE
NUMERIC("-432.876)       returns  TRUE
NUMERIC(" -432.876)      returns  TRUE
ABC="-1234.55"
NUMERIC(ABC+"44")        returns  TRUE
```

OBTAIN Statement

Gets data from an unformatted panel and assigns that input data to arithmetic and text variables. You can also use OBTAIN to retrieve data unsolicited by the panel (that is, in the lower, left corner) if the OBTAIN follows a CONVERSE.

OBTAIN *v1*, *v2*, *v3*,...

Example

```
OBTAIN ACCT_NUMBER
```

ORD Function

Use the ORD function to return the numeric value of the first character EBCDIC code.

ORD(*a*)

Example

```
ORD("C")
```

This example returns 195, which is the EBCDIC code for “C”.

OUTPUT Statement

Routes output from the CONVERSE and SHOW statements and the LIST command to a panel or printer.

$$\textbf{OUTPUT}\left\{\begin{array}{l}\textbf{SCREEN}\\ \textbf{PRINTER[VIA } \boldsymbol{e}\textbf{]}\\ \textbf{SCREEN PRINTER [VIA } \boldsymbol{e}\textbf{]}\end{array}\right\}$$

Example

```
OUTPUT SCREEN PRINTER VIA "EXAMPLEA"
```

PAD Statement

Adds occurrences of a specified character on one or both sides of a text, DBCS, or Kanji variable.

$$\textbf{PAD } \boldsymbol{v}[\boldsymbol{exp}]\begin{bmatrix}\underline{\textbf{AFTER}}\\ \textbf{ALL}\\ \textbf{BEFORE}\end{bmatrix}$$

Example

```
PAD A "*" AFTER
```

PASSWORD Function

Returns a text string containing the current password for the signed-on user.

PASSWORD

Example

```
FILE REC("CUST_FILE",PASSWORD)
```

PERFORM Statement

The PERFORM statement invokes a user-written COBOL, Assembler, or PL/I “target” program without passing parameters to it. When you perform another program, your program can either return to MANTIS or transfer control to a new program. If you are a CICS user, you can also use the PERFORM statement to run a MANTIS or external transaction as a background task. You can also end MANTIS and delete the context. Control will not resume at the statement following the PERFORM when MANTIS is invoked again. Control remains with the performed module and does not return to MANTIS.

PERFORM ***t***

Where *t* is a text expression that evaluates to:

“program”

“program/XCTL”

“program/EXEC”

“[trans-id]/BACK,user-id,password,program[;text-string]”

“trans-id/EXTN[,text.string]”

Example

```
PERFORM "/BACK,EXAMPLES,CASINO,JACKSON"
```

PI Function

Returns the value of Pi (3.141592653).

PI

Example

```
DEGREES_TO_RADIANS=PI/180
```

POINT Function

Returns a number identifying the position where the last string addition or subtraction occurs when MANTIS evaluates the text expression argument.

POINT(t1 ± t2)

Example

```
TEXT CUST_NO(11),DASH(1)
CUST_NO="123-45-6789":DASH="-"
A=POINT(CUST_NO-DASH)

A returns 4
```

PRINTER Statement

Assigns the printer device where MANTIS routes output.

PRINTER = *t*

Example

```
PRINTER=DEVICE
```

PRINTER Function

Returns the current assignment.

PRINTER

Example

```
SHOW PRINTER
```

PROGFREE Function

Returns the number of bytes remaining in the program area.

PROGFREE

Example

```
SHOW PROGFREE
```

PROGRAM Statement

Identifies an external subroutine to be invoked by a DO statement.

PROGRAM *name1*([*library1*:]*program - name1*, *password1*)
[, *name2*([*library2*:]*program - name2*, *password2*)...]

Example

```
PROGRAM EDIT_RTN("VALIDATION","COMMON")
```

PROMPT Statement

Displays a prompter. MANTIS retrieves the prompter from the library and displays it. In the case of chained prompters, MANTIS displays each prompter in the chain. Following the PROMPT, MANTIS returns control to the next line in the program.

PROMPT [*library*:]*prompter - name*

Example

```
PROMPT"MASTER:FACILITY_HELP"
```

RELEASE Statement

Frees RDM internal storage for one specific view or for all views currently opened. It also frees internal storage for programs loaded with a PROGRAM statement.

$$\textbf{RELEASE}\begin{bmatrix}\textbf{\textit{view - name}}\\ \textbf{\textit{program - name}}\end{bmatrix}$$

Example

```
RELEASE CUSTOMER
```

RELEASE Function

The RELEASE function returns a text string indicating the current release, environment, and copyright information about the MANTIS that is executing.

RELEASE

Example

```
SHOW RELEASE
```

REPLACE Statement

Identifies the library, program name, password, and description of the executable program that is replaced in your library as the result of issuing the Compose action on a MANTIS source program.

NOTE

If UPPERCASE=N has been specified in the FSE (Full-Screen Editor), you must enter this statement in UPPERCASE mode for it to be recognized by MANTIS.

REPLACE"[library:] program-name [/password] [/description]"

Example

```
REPLACE"ACCT:CUST_INSERT/DEPT1234/CUSTOMER RECORD INSERT PROGRAM"
```

RESET Statement

Backs out a Logical Unit of Work (LUW). MANTIS rolls back any updates made since the start of a logical Unit of Work. You can only back out updates when supported by the teleprocessing system and file system.

RESET

Example

```
RESET
```

RETURN Statement

Use the RETURN statement to return control from a subroutine or stop execution of a program.

RETURN

Example

The following example shows how the RETURN statement returns control from a subroutine:

```
 10 ENTRY BROWSE
 20 .SCREEN MAP1("INDEX")
 30 .FILE REC1("INDEX","SERENDIPITY")
 40 .GET REC1
 50 .WHILE REC1="NEXT"
 60 ..CONVERSE MAP1
 70 ..IF MAP1="CANCEL"<----  If this condition is TRUE, the RETURN
 80 ...RETURN                verb will continue logic
 90 ..END                    flow to the EXIT statement (120)
100 ..GET REC1
110 .END
120 EXIT
```

RND Function

Returns a random real number in the range zero to *a*, but excluding zero and *a*.

RND(*a*)

Example

```
A=INT(RND(10)+1)
```

RUN Command

Executes the program currently in the work area.

A program runs until one of the following occurs:

- The program encounters an error.
- The program encounters one of the following statements:
  - CHAIN
  - EXIT
  - RETURN
  - STOP
- The program runs out of statements.
- You issue a KILL command.

RUN[*n*]

Example

The following example shows how the RUN statement is used to execute the program currently in the work area:

```
===> RUN
00010  SHOW"WHAT IS THE CAPITAL AMOUNT?"
   WHAT IS THE CAPITAL AMOUNT?
```

SCREEN Statement

Specifies a screen design (panel) that you use in your program.

SCREEN *name1*([*library1*:]*screen - name1*,PREFIX])
[,*name2*([*library2*:]*screen - name2*,PREFIX])...]

Example

```
SCREEN MAP("INDEX")
```

SCROLL Statement

Sets the scrolling mode of the terminal or specifies (within your program) window-mode scrolling increments for PF keys.

$$\textbf{SCROLL}\begin{Bmatrix}\textbf{OFF}\\ \underline{\textbf{ON}}\\ \textbf{[}\textit{\textbf{row}}\textbf{][,}\textit{\textbf{col}}\textbf{]}\end{Bmatrix}$$

Example

```
SCROLL OFF
```

SEED Statement

“Seeds” the random number generator so that it generates a new sequence of random numbers.

SEED

Example

```
SEED
```

SGN Function

Returns the algebraic sign of a numeric expression:

- -1 if *a* is less than 0
- 0 if *a* equals 0
- +1 if *a* is greater than 0

SGN(*a*)

Example

```
SGN(-14)   returns –1
```

SHOW Statement

Displays and formats data on an unformatted panel. MANTIS outputs the specified data item(s) on the panel according to the scrolling method specified. MANTIS locates these data items on the line according to the AT, ‘,’ and ‘;’ options specified.

$$\textbf{SHOW}\quad \boldsymbol{x1}\left[\begin{Bmatrix} , \\ ; \end{Bmatrix}\ \boldsymbol{x2}\begin{Bmatrix} , \\ ; \end{Bmatrix}\ \ldots\boldsymbol{xn}\ \begin{Bmatrix} , \\ ; \end{Bmatrix}\right]$$

$$\text{where each } \boldsymbol{xn} \textbf{ is :}\left[\ \textbf{AT(}\boldsymbol{tab}\textbf{)}\quad \begin{Bmatrix} , \\ ; \end{Bmatrix}\quad \boldsymbol{data - item}\right]$$

Example

```
SHOW NUMBER,LAST_NAME,FIRST_NAME
```

SIN Function

Returns the sine of *a* where *a* is in radians.

SIN(*a*)

Example

```
SHOW SIN(100)
returns: -.506365641
```

SIZE Function

Returns the size, and the maximum or current length of a field. It can also return the number of defined dimensions for a field or array as well as the number of occurrences for a specific dimension of an array.

$$\textbf{SIZE (}\textbf{\textit{field - name}}\begin{bmatrix} \textbf{,"MAX"} \\ \textbf{,"DIM"} \\ \textbf{,}\textbf{\textit{n}} \\ \textbf{,"BYTE}\textbf{\textit{length}}\textbf{"} \end{bmatrix}\textbf{)}$$

SIZE(*text-expression*)

Example

```
A=SIZE(CUST_NAME,"DIM")
```

SLICE Statement

Limits the number of statements you can execute before MANTIS suspends your program.

SLICE *n*

SLICE CLEAR

Example

```
SLICE 1000
```

SLOT Statement

Specifies how many times a program can reach the SLICE limit before MANTIS returns “POTENTIAL PROGRAM LOOP ENCOUNTERED”. You can press ENTER to continue running the program, or use the KILL command to terminate execution.

SLOT *n*

Example

```
SLOT 20
```

SMALL Statement

Names and gives dimensions to numeric variables. MANTIS creates a four-byte numeric floating-point field or an array of four-byte fields and associates it with the specified name.

SMALL *name1*[(*n1*[, *n2*])[, *name2*[(*n1*[, *n2*]) . . .]

Example

```
SMALL ALPHA(64,3),BETA(12)
```

SOURCE Statement

Coded in an executable program to name the library, program, password, and description of the source program to be created or replaced in your library by the Decompose action.

NOTE

If UPPERCASE=N has been specified in the FSE (Full-Screen Editor), you must enter this statement in UPPERCASE mode for it to be recognized by MANTIS.

SOURCE"[*library*:] *program - name* [*/password*] [*/description*]"

Example

```
|@SOURCE"ACCT:CUST_INSERT@/DEPT1234/CUSTOMER RECORD INSERT"
```

SQLCA Statement

Stores data from the MANTIS program into the SQL Communication Area (SQLCA).

SQLCA(*sqlca_ element_ name*) = *expression*

Example

```
SQLCA("SQLCODE")=-504
```

SQLCA Function

Transfers data from the SQL Communication Area (SQLCA) into the MANTIS program.

SQLCA(*sqlca_ element_ name*)

Example

```
SQLCA("SQLCODE)=100
```

SQLDA Statement

The SQLDA statement (write) stores data from the MANTIS program into the SQL Descriptor Area (SQLDA).

$$\textbf{SQLDA(}\textbf{\textit{sqlda\_name}}\textbf{) =} \left\{\begin{array}{l}\textbf{NEW}\\ \textbf{QUIT}\end{array}\right\}$$

$$\textbf{SQLDA(}\textbf{\textit{sqlda\_name}}\textbf{,}\left\{\begin{array}{l}\textbf{\textit{sqlda\_header\_element}}\textbf{)}\\ \textbf{\textit{repeating\_element, index}}\textbf{)}\end{array}\right\}\textbf{ = }\textbf{\textit{expression}}$$

Example

```
SQLDA("SQLDA1")=NEW
```

SQLDA Function

The SQLDA function (read) transfers data from the SQLDA into the MANTIS program.

$$\textbf{SQLDA(}\textbf{\textit{sqlda\_name}}\textbf{,}\left\{\begin{array}{l}\textbf{\textit{sqlda\_header\_element}}\textbf{)}\\ \textbf{\textit{sqlda\_repeating\_element, index}}\textbf{)}\end{array}\right\}$$

Example

```
IF SQLDA("SQLDA1","SQLN")<4
.SQLDA("SQLDA1","SQLN")=4
END
```

SQR Function

Returns the square root of a non-negative arithmetic expression.

SQR(*a*)

Example

```
SHOW SQR(100)
returns:  10
```

STOP Statement

Terminates program execution. When a STOP statement is executed, MANTIS:

- Returns to programming mode if the program was executing while in programming mode.

- Returns to your Facility Selection menu if the program was executing while not in programming mode.

STOP

Example

```
STOP
```

TAN Function

Returns the tangent of *a*, where *a* is a valid numeric value in radians.

TAN(*a*)

Example

```
TAN(10)
```

This example returns .648360827.

TERMINAL Function

Returns a text string of 1-8 characters containing the terminal ID.

TERMINAL

Example

```
IF TERMINAL="XX02"
.SHOW TERMINAL
.SHOW "NOT AUTHORIZED FROM THIS TERMINAL":WAIT
END
```

TERMSIZE Function

Returns the size of the current terminal in rows and columns.

TERMSIZE

Example

```
SHOW TERMSIZE
```

TEXT Statement

Names and specifies dimensions for text variables and lists.

```
           ⎡   [n,]length  ⎤⎡        [n,]length   ⎤    ⎤
TEXT name1 ⎢ (             )⎥⎢ ,name2(             ) ⎥... ⎥
           ⎣   16          ⎦⎣        16           ⎦    ⎦
```

Example

```
TEXT ALPHA
```

TIME Statement

Specifies a text string by which the TIME function formats the current time.

TIME=*mask-expression*

Example

```
TIME="HH:MM AM"
SHOW TIME
1:17 PM
12:34 AM
```

TIME Function

Returns a text string containing the current time in the format of the current specification.

TIME

Example

```
SHOW TIME
11:21:05
```

TOTAL Statement (TOTAL and SUPRA PDM users only)

Specifies a TOTAL file view. MANTIS retrieves the specified TOTAL file view from your library and validates it, for consistency with the active Data Base Descriptor Module (DBMOD) or SUPRA directory. If there is an inconsistency, MANTIS returns an error message and halts execution.

If TOTAL variables are not already defined in your program, MANTIS defines them. If TOTAL variables are already defined, MANTIS checks for consistency. ON or OFF issues a PDM SINON or SINOF function.

TOTAL {
***name1*([*library1*:]TOTAL – *view1*, *password1*[, PREFIX][, *n1*])**
[, *name2*([*library2*:]TOTAL - *view2*, *password2*[, PREFIX][, *n2*])...]
ON
OFF
}

Example

```
TOTAL CUSTOMER("CLIENT","SALES")
```

TRAP Statement

Intercepts I/O errors from MANTIS files, external VSAM files, or additional status codes returned by TOTAL or RDM. The program continues to execute for trapped errors. With TRAP ON, MANTIS reflects the status of the last GET, UPDATE, DELETE, or INSERT statement in the value of a symbolic variable *name*.

TRAP *name* $\left[\begin{array}{l}\textbf{ON}\\ \underline{\textbf{OFF}}\end{array}\right]$

Example

```
TRAP REC ON
```

TRUE Function

Returns a value of +1 and can be used to set conditions within your program.

TRUE

Example

```
ERROR_OCCURRED=TRUE
```

TXT Function

Returns the text value of a numeric expression, *a*.

TXT(*a*)

Example

```
KEY_TO_PRESS = "PF" + TXT(OPTION_NUMBER)
```

UNPAD Statement

Removes all occurrences of a specified character from either or both sides of a text, DBCS, or Kanji variable.

$$\textbf{UNPAD } \textbf{\textit{v}} \textbf{ [}\textbf{\textit{exp}}\textbf{]} \begin{bmatrix} \textbf{BEFORE} \\ \underline{\textbf{AFTER}} \\ \textbf{ALL} \end{bmatrix}$$

Example

```
UNPAD CLIENT_NAME
```

UNTIL-END Statement

Executes a block of statements repeatedly until a specified condition becomes true. MANTIS executes the block of statements in the range of the UNTIL once before it tests the condition.

UNTIL *expression*
        .
        . ***statements***
        .
END

Example

```
UNTIL KEY="CANCEL"
.SHOW RND(10)
.WAIT
END
```

UPDATE Statement

Replaces a record on a MANTIS file, an external file, a personal computer file with an updated (altered) record, an RDM logical view, or a TOTAL DBMS view. You do not need to read a MANTIS record before updating it. MANTIS returns a text string, in either the *file-name* or *view-name* variable, that reflects the operation's status.

MANTIS file, External file, PC file, TOTAL file

UPDATE *file - name* [LEVEL = *n*]

RDM Logical view

UPDATE *view - name* [LEVEL = *n*]

Example

```
UPDATE RECORD LEVEL=COUNTER
```

UPPERCASE Function

Converts a text string into uppercase.

UPPERCASE(*t*)

Example

```
IF UPPERCASE (T1)=UPPERCASE(T2)
```

USAGE Command

Determines where a symbolic name appears in a program. MANTIS searches each statement within a given range for the symbolic name you specify. USAGE is valid only in the MANTIS line editor, and only for MANTIS symbolic names.

USAGE *name* [,*starting_line_number* [,*occurrences_to_display*]]

Example

```
USAGE  REC1
FILE  REC1("INDEX","SERENDIPITY")
GET  REC1
WHILE REC1<>"END"
.GET  REC1
   .
   .
   .
```

USER Function

Returns a text string identifying the current user name.

USER

Example

```
HEAD"ENTITY "+ENTITY+" FOR USER "+USER
```

USERWORDS Function

Returns the number of MANTIS symbolic names currently in use.

USERWORDS

Example

```
SHOW USERWORDS
```

VALUE Function

Returns the numeric value of a text expression.

VALUE(*t*)

Example

```
WHEN VALUE(CMD(1,5))>ZERO AND VALUE(CMD(1,5))<99999
```

VIEW Statement (SUPRA RDM users only)

Specifies an RDM logical view. MANTIS retrieves the view from the SUPRA Directory. If the view is not known to RDM or not authorized for your use, MANTIS returns an error message and halts execution. If VIEW variables are not already defined in your program, MANTIS defines them. If VIEW variables are already defined, MANTIS checks for consistency.

If the view is valid, MANTIS opens it and establishes the MANTIS variables as they are defined in the SUPRA Directory with two exceptions:

- MANTIS converts all hyphens (-) in logical view field names to underscores (_) in MANTIS variable names and vice versa.
- The characters $ and # are invalid in MANTIS. If the logical view has field names with these characters in them, MANTIS returns an error message and halts execution.

```
     ⎧name1(view-name1[,PREFIX][,n1]
     ⎪                                [,SELECT (f1list1[,f1list2][,...][,f1listn])])⎫
     ⎪         [,name2(view-name2[,PREFIX][,n2]                                        ⎪
VIEW ⎨                                [,SELECT (f2list1[,f2list2][,...][,f2listn])])⎬
     ⎪         ...                                                                     ⎪
     ⎪ON[user-id[,password]])]                                                         ⎪
     ⎩OFF                                                                              ⎭
```

Example

```
VIEW CUST_ITEM("CUST_ITEM",10,SELECT("CUST_NO,ITEM_NUM"))
```

VSI Function

Indicates the highest field status for the last operation on a view.

VSI(*view - name*)

Example

```
IF VSI(PARTS)="CHANGED"
```

WAIT Statement

Temporarily suspends execution of a program. Generally, use the WAIT statement to display unformatted data (from a SHOW statement) on the panel until you press ENTER to continue execution. You must use a WAIT statement to display SHOWS when you execute a program from a menu.

WAIT

Example

```
SHOW REC:WAIT
```

WHEN-END Statement

Executes a block of statements only if a specified condition is met. MANTIS performs the test before executing the block. If the condition is false, or after executing the block, execution proceeds to the next WHEN.

WHEN *expression*
. *statements*
WHEN *expression*
. *statements*
END

Example

```
..WHEN MAP="PF3"
...UPDATE RECORD
..END
```

WHILE-END Statement

Executes a block of statements repeatedly while a specified condition is true. If the condition is FALSE, MANTIS terminates the WHILE and executes the statement after END. If the condition is TRUE, MANTIS executes the statements within the WHILE range and then reevaluates the relational expression.

WHILE *expression*

.statements

END

Example

```
.WHILE RECORD<>"END"AND MAP<>"CANCEL"
..CONVERSE MAP
..WHEN MAP="PF1"
...INSERT RECORD
..WHEN MAP="PF2"
...DELETE RECORD
..WHEN MAP="PF3"
...UPDATE RECORD
..END
..GET RECORD
.END
```

ZERO Function

Returns the value zero.

ZERO

Example

```
NATION_COUNTER=ZERO
```

4

Built-in functions and constants

Overview of built-in functions and constants

MANTIS contains numeric and text functions used to return values within a program. The following table lists and describes these functions in alphabetical order and provides information on the input and output used by each function and the function type. In the descriptions, *a* represents any arithmetic expression; *k* represents any Kanji expression; *t* represents any text expression.

Function	Description	Input	Output	Type of function
ABS (*a*)	Returns the absolute value of *a*.	Numeric	Numeric	Mathematical
ASI	Indicates the status of a field in a logical view.	Field-name	Text	File Access
ATN (*a*)	Returns the angle in radians whose tangent is *a.*	Numeric	Numeric	Mathematical
ATTRIBUTE	Returns the current status of field, map, terminal, and printer attributes.	Name or Reserved word	Text	System
CHR (*a*)	Returns the EBCDIC character of the numeric value *a*.	Numeric	Text	String
COS (*a*)	Returns the cosine of *a* where *a* is in radians.	Numeric	Numeric	Mathematical

Function	Description	Input	Output	Type of function
CURSOR	Indicates whether cursor appeared in a specific field at the last terminal I/O.	Field-name	True/False	System
DATAFREE	Returns the number of bytes remaining in the data area.	None	Numeric	System
DATE	Returns a text character string of the current date.	None	Text	System
DOLEVEL	Returns your current level in an external subroutine.	None	Numeric	System
E	Returns the value of natural E (2.71828182845905) .	None	Numeric	Mathematical
EXP (*a*)	Returns the value of natural E to the power of *a*.	Numeric	Numeric	Mathematical
FALSE	Returns the value zero.	None	Numeric	Boolean
FORMAT	Returns a text-string conversion of a numeric expression according to a supplied edit mask.	Numeric	Text	System
FSI	Indicates the status of a file after an I/O (GET, DELETE, INSERT, or UPDATE).	File-name	Text	File Access
INT (*a*)	Returns the integer value of *a*.	Numeric	Numeric	Mathematical
KEY	Returns text character string reflecting the last key pressed in response to a CONVERSE, OBTAIN, PROMPT, or WAIT statement.	None	Text	System

Function	Description	Input	Output	Type of function
LANGUAGE	Returns the current language code.	None	Text	System
LOG (*a*)	Returns the natural logarithm of *a*.	Numeric	Numeric	Mathematical
LOWERCASE (*t*)	Converts a text expression into lowercase.	Text	Text	String
LUID	Returns an 8-character text string containing the VTAM logical unit ID.	None	Text	System
MIXD (*t*)	Retrieves DBCS data from mixed data.	Text	DBCS	String
MIXM (*k*)	Returns a mixed data string containing shift codes from DBCS.	DBCS	Text	String
MIXT (*t*)	Retrieves text (SBCS) string from mixed data.	Text	Text	String
MODIFIED	Tests whether a specified field or number of fields within a map definition changed during the last physical I/O.	Map/ Field-name	Numeric	System
NOT (*a*)	Returns TRUE(1) if *a* evaluates to FALSE(0). Otherwise, returns FALSE(0).	Numeric	Numeric	Boolean
NULL	Returns a zero length text string.	None	Text	String
NUMERIC (*a*)	Returns TRUE (1) if string *a* contains only a valid numeric value. Otherwise, returns FALSE.	Field-name	Numeric	Boolean

Function	Description	Input	Output	Type of function
ORD (*t*)	Returns the numeric value of the first EBCDIC character in *t*.	Text	Numeric	String
PAD (*statement*)	Fills in either or both sides of a Kanji or text variable with a specified character.	Text or DBCS	Text or DBCS	String
PASSWORD	Returns a text character string containing the current password.	None	Text	System
PI	Returns the value of Pi (3.14159265358979) .	None	Numeric	Mathematical
POINT (*t* ± *t*) (*k* ± *k*)	Returns a number representing the position where a string addition or subtraction would occur if you executed it.	Text or DBCS	Numeric	String
PRINTER	Returns a text character string containing the current printer assignment.	None	Text	System
PROGFREE	Returns the number of bytes remaining in the program area.	None	Numeric	System
RND (*a*)	Returns a random real number in the range zero to *a*, but excluding zero and *a*.	Numeric	Numeric	Mathematical
SGN (*a*)	Returns the algebraic sign depending on the value of *a.*	Numeric	Numeric	Mathematical

Function	Description	Input	Output	Type of function
SIN (*a*)	Returns the sine of *a*, where *a* is in radians.	Numeric	Numeric	Mathematical
SIZE	Returns the size, dimensions, or byte length of a field.	Text or DBCS	Numeric	String
SQLCA	Transfers data between the MANTIS program and the SQL Communications Area.	Varies	Varies	SQL
SQLDA	Allows MANTIS programs to access an SQL Descriptor Area.	Varies	Varies	SQL
SQR (*a*)	Returns the square root of *a*.	Numeric	Numeric	Mathematical
TAN (*a*)	Returns the tangent of *a* where *a* is in radians.	Numeric	Numeric	Mathematical
TERMINAL	Returns a text character string of 1-8 characters containing the terminal ID.	None	Text	System
TERMSIZE	Returns terminal size in rows and columns.	None	Text	System
TIME	Returns a text character string of the current system time.	None	Text	System
TRUE	Returns the value + 1.	None	Numeric	Boolean
TXT (*a*)	Returns the text value of *a*, up to nine significant digits.	Numeric	Text	String

Function	Description	Input	Output	Type of function
UNPAD (*statement*)	Removes the extra specified characters from one or both sides of a Kanji or text variable.	Text or DBCS	Text or DBCS	String
UPPERCASE (*t*)	Converts a text expression into uppercase.	Text	Text	String
USER	Returns a text character string containing the current user name.	None	Text	System
USERWORDS	Returns the number of MANTIS symbolic names currently in use.	None	Numeric	System
VALUE (*t*)	Returns the numeric value of the text string *t.*	Text	Numeric	String
VSI	Indicates the highest field status within a logical record following a terminal I/O.	File-name	Text	File Access or System
ZERO	Returns the value zero.	None	Numeric	Mathematical

5

Special characters

Overview of special characters

The following table lists MANTIS special characters and describes how they are used.

Character	Description
#	Hash character. Designates data fields in Screen Design. May be user-defined.
“ ”	Double quotes. Encloses a text literal (may be different in some countries). May be user-defined.
'	Single quote (apostrophe). Signifies a continuation line in programming mode.
( )	Parentheses. Used in arithmetic or text expressions and in the FILE, SCREEN, and INTERFACE and other library statements for naming conventions.
:	Colon. Separates two programming statements on the same line.
;	Semicolon. Indicates tabbing on an unformatted panel, as explained in the SHOW statement. Also separates parameters.
,	Comma. Separates parameters and subscripts, and indicates tabbing on an unformatted panel.
.	Period. Designates a decimal point in a number. May be user-defined.
_	Underline. Connects two or more words in a symbolic name.

<table>
<tr><th>Character</th><th>Description</th></tr>
<tr><td>|</td><td>Vertical bar. Marks a comment line in programming mode. In Screen Design, it is the default blank-fill character. May be user-defined for screen design. It joins fields (e.g., words in a heading) or indicates automatic skipping (tabbing) between fields.</td></tr>
<tr><td>!</td><td>Exclamation point. Marks a Kanji or Double Byte Character String (DBCS) comment.</td></tr>
<tr><td>+</td><td>Plus sign. Adds two data items.</td></tr>
<tr><td>-</td><td>Minus sign. Subtracts two or more data items. Don’t use a minus sign (or dash) between two words in a file name (e.g., file-name) because MANTIS will try to subtract the names.</td></tr>
<tr><td>*</td><td>Asterisk. Multiplies two or more data items.</td></tr>
<tr><td>**</td><td>Double asterisk. Raises one number to the power of a second number (exponentiation).</td></tr>
<tr><td>></td><td>Greater than sign. Used as a comparison operator in > (greater than), <> (not equal) and >= (greater than or equal to).</td></tr>
<tr><td><</td><td>Less than sign. Used as a comparison operator in < (less than), <> (not equal) and <= (less than or equal to).</td></tr>
</table>

Character	Description
/	Slash. Divides one number by the value of a second number.
=	Equal sign. ♦ Comparison operator evaluates an expression to TRUE if both sides are equal. Otherwise, it evaluates the expression to FALSE. ♦ Assignment operator in a LET statement sets the variable on the left-hand side to the value of the expression(s) on the right-hand side. ♦ Used for ATTRIBUTE, PRINTER, DATE, LANGUAGE, SQLCA, SQLDA, TERMSIZE, and TIME statement forms. ♦ Used in LEVEL= specification of I/O statements. ♦ Used for initial value of FOR statement. ♦ Used for keywords of COMPONENT, CSIOPTNS, and SOURCE statements.
@	At sign. Used by the Component Engineering Facility to recognize a source program when found as suffix to a MANTIS program name. Also used to “nominate” the Decompose process. May be user-defined.

6

Operators

MANTIS evaluates operators in the following order:

( )	Expressions in parentheses
Unary +, Unary -	Unary operators
**	Exponentiation (to the power of)
*, /	Multiplication, Division
+, -	Addition, Subtraction
>, <, =, > =, < =, < >	Relational Operators
AND	Conjunction
OR	Disjunction

NOTE

You can use parentheses ( ) to change the order of operation.

7

Attribute types

Overview of attribute types

The following table lists, by type, attribute values that you can set.

NOTE

For a list of the values returned by the ATTRIBUTE function, see “ATTRIBUTE function (values returned)” on page 131.

NOTE

Enter only the underlined portions of the words listed in the Attribute column.

Attribute	Terminal	Printer	Cursor	Screen	Field	Screen design
AUTO SKIP/NO AUTO SKIP				✓	✓	✓
BLINK/NO BLINK	✓			✓	✓	✓
BOXED/UNBOXED	✓	✓		✓	✓	✓
BRIGHT / NORMAL / HIDDEN	✓			✓	✓	✓
CLASS		✓				
COLOR/NO COLOR	✓	✓				
CURSOR / NOCURSOR				✓	✓	✓

Attribute	Terminal	Printer	Cursor	Screen	Field	Screen design
DETECTABLE / NON DETECTABLE	✓	✓		✓	✓	✓
DEFAULT / RANGE / FILL / MASK / REQUIRED						✓
FULL DISPLAY / NO FULL DISPLAY				✓		✓
HIGHLIGHT / NO HIGHLIGHT				✓	✓	✓
KANJI/NOKANJI	✓	✓				✓
KEEP MAP MODIFIED / RESET MAP MODIFIED				✓		
LEFT BAR / NO LEFT BAR	✓	✓		✓	✓	✓
MIX / NOMIX	✓			✓	✓	✓
MODIFIED / UNMODIFIED				✓	✓	✓
NATIVE LANGUAGE SUPPORT	✓					
NUMERIC / TEXT / KANJI / HEADING						✓

Attribute	Terminal	Printer	Cursor	Screen	Field	Screen design
NO COLOR / NEUTRAL / BLUE / PINK / GREEN / TURQUOISE / RED / YELLOW				✓	✓	✓
OVERLINE / NO OVERLINE	✓	✓		✓	✓	✓
PROTECT BOTTOM LINE / BOTTOM LINE ENTRABLE				✓		✓
PROTECTED / UNPROTECTED				✓	✓	✓
RESET				✓	✓	
REVERSE VIDEO / VIDEO	✓	✓		✓	✓	✓
RIGHT BAR / NO RIGHT BAR	✓	✓		✓	✓	✓
SEND ALL FIELDS / SEND MODIFIED FIELDS	✓					
SOUND ALARM / NO ALARM				✓		✓
UNDERLINE / NO UNDERLINE	✓	✓		✓	✓	✓
UPPERCASE / LOWERCASE	✓	✓		✓	✓	✓

8

ATTRIBUTE function (values returned)

Below is an alphabetical list of the values that can be returned by the ATTRIBUTE function. The underlining indicates the abbreviation MANTIS uses to return the value. For example, “BLI” replaces a field that is set to “BLINK”.

- AUTO SKIP
- BLINK
- BOXED
- BRIGHT
- COLOR
  - BLUE
  - RED
  - PINK
  - GREEN
  - TURQUOISE
  - YELLOW
  - NEUTRAL
  - NO COLOR
- CURSOR
- DEFAULT VALUE
- DETECTABLE
- DEVICE CLASS
- FILL
- FULL DISPLAY
- HEADING
- HIDDEN
- HIGHLIGHT
- INTERNAL MIXED
- KANJI
- KEEP MAP MODIFIED
- MIXED - SO/SI
- MODIFIED
- NLS
- NORMAL
- NUMERIC
- NUMERIC MASK
- OVERLINE
- PROTECT BOTTOM LINE
- PROTECTED
- RANGE CHECK
- RIGHT BAR
- REQUIRED
- REVERSE VIDEO
- SEND ALL FIELDS
- SOUND ALARM
- TEXT
- UNDERLINE
- UNDERLINE
- UPPERCASE

9

Screen Design PF keys

Create or Update a Screen

Key	Function
PF1/13	Inserts a line.
PF2/14	Deletes a line.
PF3/15	Displays/removes the column scale line.
PF4/16	Moves a field.
PF5/17	Copies a field.
PF6/18	Deletes a field.
PF7/19	Scrolls the screen up.
PF8/20	Scrolls the screen down.
PF9/21	Displays/removes the row scale line.
PF10/22	Scrolls the screen left.
PF11/23	Scrolls the screen right.
PF12/24	Returns window to its origin (row 1, column 1).

Update Field Specifications/Update Repeat Specifications

Key	Function
PF1/13	Selects the first field on the screen.
PF2/14	Selects the next field (field following the last processed field) on the screen.
PF3/15	Presents undefined fields for definition one at a time.
PF4/16	Selects the last field on the screen.
PF5/17	Selects the previous field (field before the last processed field).
PF6/18	Selects fields from a field table.
PF7/19	Presents a range of fields to be processed one at a time. Position cursor in first field of range and press PF7/19; position cursor in last field of range and press PF7/19 again. Each field is then presented in sequence for update.
PF8/20	Swaps between defining field specifications and defining repeat specifications.

Screen Display

Key	Function
PF6/18	Returns to Screen Design Facility menu.
PF7/19	Scrolls the screen up.
PF8/20	Scrolls the screen down.
PF9/21	Terminates window mode.
PF10/22	Scrolls the screen left.
PF11/23	Scrolls the screen right.
PF12/24	Returns to origin (row 1, column 1).

10

Screen Design commands

L1 is the starting line number; L2 is the ending line number; L3 is the destination line number; n is the specified number of lines.

Command	Description
CLEAR	Clears current screen design from work area.
COPY L1[,L2],L3	Copies a single line or a range of lines.
DEFAULTS	Displays the default row and column domains and allows updates for scrolling amount and blank fill character.
DELETE L1[,L2]	Deletes a single line or a range of lines.
HELP	Displays help prompters for PF keys and commands.
INSERT L1[,n]	Inserts a single line or a range of lines.
MOVE L1[,L2],L3	Moves a single line or a range of lines.
NEW	Clears the current screen design from the work area.

11

Search Facility

The MANTIS Search Facility (MSF) is a programmer’s tool to search through the text of MANTIS entities and find matches for supplied criteria.

Accessing the MANTIS Search Facility

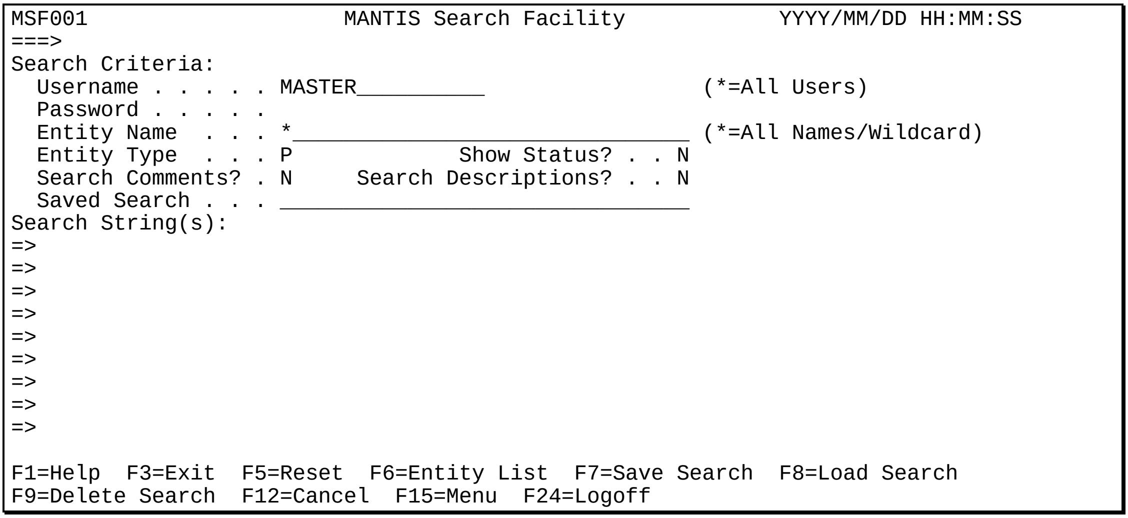
The MANTIS Search Facility is executed using online or Batch MANTIS. Sign on to the appropriate user desired, and select Search Facility.

NOTE

You can run this facility online or in batch, and it can produce a report of matches found.

Search Facility screen

When you select the Search Facility, or you run it by its name (CONTROL: SEARCH_FACILITY), the screen appears as shown in the following screen:



```
MSF001                    MANTIS Search Facility            YYYY/MM/DD HH:MM:SS
===>
Search Criteria:
  Username . . . . . MASTER__________                (*=All Users)
  Password . . . . .
  Entity Name  . . . *_______________________________ (*=All Names/Wildcard)
  Entity Type  . . . P            Show Status? . . N
  Search Comments? . N     Search Descriptions? . . N
  Saved Search . . . ________________________________
Search String(s):
=>
=>
=>
=>
=>
=>
=>
=>
=>

F1=Help  F3=Exit  F5=Reset  F6=Entity List  F7=Save Search  F8=Load Search
F9=Delete Search  F12=Cancel  F15=Menu  F24=Logoff
```

Field descriptions for the Search Facility screen are listed below:

username

Description The user name defaults to whatever user is signed on. You may change the user name to any other valid MANTIS user.

Consideration Exceptions include:

- Control users (users with usercodes less than 16) cannot be accessed.
- The Master User can only be accessed by Master.
- The Master User can enter a wildcard (*) to search across all users.

password

Description If the signed on user is not Master and a different user is specified, the password for that user must also be entered.

The Master User may enter any other user name without required to enter the password for that user.

entity name

Description The entity name is the name of the entity you want to search

Options A complete valid entity name—searches only that entity.

Considerations

- A partial name with an asterisk (*) as a wildcard—searches all entities beginning with the partial name. (For example, CUST* searches all entities beginning with “CUST”.)
- An asterisk (*)—searches all entity names.

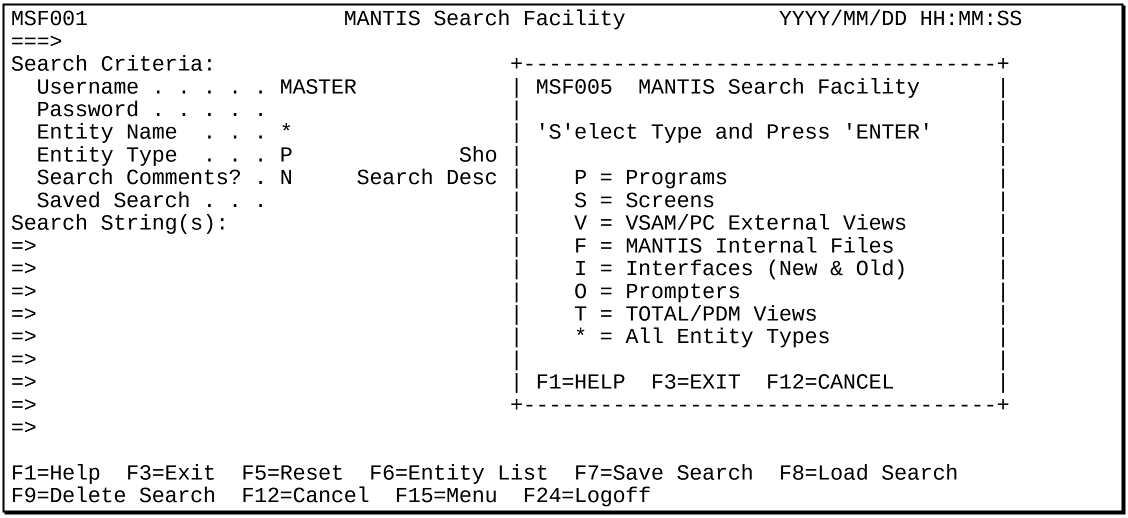
entity type

Description The entity type can be one or more one-character letters representing the different types of entities to be searched.

Options

P	Programs (default)
S	Screens
F	MANTIS Internal Files
I	Interfaces (both old and new-style)
O	Prompters
V	VSAM/PC/Access External File Views
T	TOTAL File Views
*	All entity types

You can also press PF6 to display an entity type selection screen, shown in the following screen:



```
MSF001                    MANTIS Search Facility          YYYY/MM/DD HH:MM:SS
===>
Search Criteria:                       +-------------------------------------+
  Username . . . . . MASTER            | MSF005  MANTIS Search Facility      |
  Password . . . . .                   |                                     |
  Entity Name  . . . *                 | 'S'elect Type and Press 'ENTER'     |
  Entity Type  . . . P             Sho |                                     |
  Search Comments? . N     Search Desc |    P = Programs                     |
  Saved Search . . .                   |    S = Screens                      |
Search String(s):                      |    V = VSAM/PC External Views       |
=>                                     |    F = MANTIS Internal Files        |
=>                                     |    I = Interfaces (New & Old)       |
=>                                     |    O = Prompters                    |
=>                                     |    T = TOTAL/PDM Views              |
=>                                     |    * = All Entity Types             |
=>                                     |                                     |
=>                                     | F1=HELP  F3=EXIT  F12=CANCEL        |
=>                                     +-------------------------------------+
=>

F1=Help  F3=Exit  F5=Reset  F6=Entity List  F7=Save Search  F8=Load Search
F9=Delete Search  F12=Cancel  F15=Menu  F24=Logoff
```

Place an “S” next to the type(s) of searches you want and press ENTER to fill the Entity Type field on the MSF panel.

Show status

Description

When a large scope search is conducted online, it may appear that the MANTIS Search Facility is hanging up. Searching hundreds of programs with thousands of lines of code can cause delays as matches are found. By changing the Show Status indicator, you can receive feedback during the search.

- Entering “S” indicates summary feedback. After 25 or more entities are searched without a match found, a message displays to the screen.

```
Scanning Programs ...
MSFSUMI:Scanned 25 entities without finding match - ENTER to continue search
```

Pressing ENTER will continue the search. You can optionally enter KILL (or whatever your Master User has designated) to abort the search.

- Entering “D” indicates detailed feedback. Every entity where no match is found will be listed to the screen.

```
Scanned Program MASTER:ADD_TO_MENU ...
Scanned Program MASTER:ADV_ANALYZE_LOG ...
Scanned Program MASTER:ADV_DUMMY ...
Scanned Program MASTER:ADV_SIGN_ON ...
Scanned Program MASTER:ADV_SQL_CONNECT …
Scanned Program MASTER:ADV_SYS_EXTERNAL ...
…
```

Pressing ENTER will continue the search. You can optionally enter KILL (or whatever your Master User has designated) to abort the search.

- Entering “N” (default) indicates no feedback. MANTIS Search Facility will only display the results screen.

Search comments

- Entering “Y” indicates that while searching programs, comments are to be included in the search.
- Entering “N” (default) excludes comments from the search.

Search description

- Entering a “Y” indicates that entity descriptions are to be included in the search.
- Entering “N” (default) excludes entity descriptions from the search.

Saved search

- You have the ability to save and recall search criteria. This is discussed in the following section.

Search string(s)

- You may enter up to nine separate search criteria. Each search criterion is examined and compared separately as the search is conducted. If a string within an entity matches one or more of the search strings, it is included in the results display.

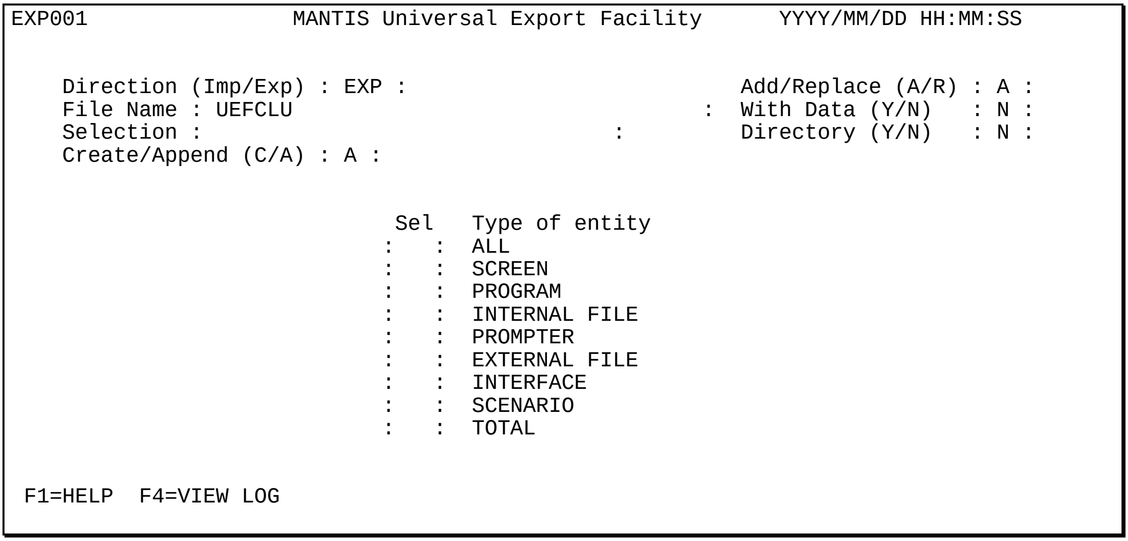
12

Universal Export Facility (UEF)

The Universal Export Facility (UEF) enables you to export or import one or more entities (such as screens, files, programs, and views) from one MANTIS cluster to another. The MANTIS clusters can be on the same or different platforms.

Universal Export Facility menu

When you select the Universal Export Facility Option from the MANTIS Master Facility Selection Menu, the Universal Export Facility menu is displayed, as shown in the following screen illustration:



```
EXP001                MANTIS Universal Export Facility      YYYY/MM/DD HH:MM:SS


    Direction (Imp/Exp) : EXP :                          Add/Replace (A/R) : A :
    File Name : UEFCLU                                :  With Data (Y/N)   : N :
    Selection :                                :         Directory (Y/N)   : N :
    Create/Append (C/A) : A :


                              Sel   Type of entity
                             :   :  ALL
                             :   :  SCREEN
                             :   :  PROGRAM
                             :   :  INTERNAL FILE
                             :   :  PROMPTER
                             :   :  EXTERNAL FILE
                             :   :  INTERFACE
                             :   :  SCENARIO
                             :   :  TOTAL


 F1=HELP  F4=VIEW LOG
```

Direction (Imp/Exp)

Description *Required*. Specifies the direction of transfer.

Options IMP Imports entities into the MANTIS file from an external file

EXP Exports entities from the MANTIS file to an external file

Add/Replace (A/R)

Description *Optional*. Specifies whether to Add or Replace the entities in the MANTIS file.

Default A

Options A Adds entities to the MANTIS file if they do not already exist. If they do exist in the MANTIS file, an error displays.

R Replaces entities in the MANTIS file if they already exist; adds entities to the MANTIS file if they do not exist.

Considerations

- This field is used only for IMPORT.
- If you select the option WITH DATA when replacing an existing internal file view, associated data is also replaced by the data in the export file. If you do not select the option WITH DATA when replacing, associated data is lost.

File Name

Description *Required*. Specifies the external sequential file from which (IMPORT) or to which (EXPORT) you want the entity transferred.

With Data

Description *Optional*. Specifies whether to import/export the data of the internal file entities.

Default N No

Options N No Import/export file entity without data

Y Yes Import/export file entity with data

Consideration This field applies to internal files only.

Selection

Description *Optional*. Specifies the name of the single entity to import/export, for the type you select in the SEL field.

Format An entity name or a wild card expression

Considerations

- ♦ Entering nothing imports/exports all entities of the selected type (SEL field).
- ♦ Specifying N for the field DIRECTORY exports all entities specified here. When you specify Y for the field DIRECTORY, you can select entities to export from the displayed portion of the directory.

Directory

Description *Optional*. Specifies whether to display the directory of the selected entity type. From the directory, you can select entities to export.

Default N No

Options N No Does not display directory

Y Yes Displays directory

Considerations

- ♦ This field is used only for EXPORT.
- ♦ Entering “Y” here and an entity name in the field SELECTION displays the part of the directory beginning with the specified name. However, entering a wild card expression in the field SELECTION displays only the relevant entities.

Create/Append (C/A)

Description *Required*. Indicates whether entities are placed in a new external text file or appended to an existing external text file.

Default A Append

Options A Append to the existing export file

C Create new external text file (not available on IBM mainframe)

Sel

Description *Required*. Selects the type(s) of entity to be imported/exported.

Format S

Considerations

- You must select at least one of the types or “ALL”.
- If you select “ALL”, you cannot select any other types.
- Selecting “ALL” corresponds to selecting programs, screens, prompters, internal files, scenarios, interfaces, TOTAL views and external files.

Message line

Description *Display*. When transfer of entities is complete or terminates, the Message Line displays the following informational messages:

Number of entities exported/imported

Number of data records exported/imported

Number of errors

Number of warnings

F1=HELP

Description *Display*. Press F1 to view a series of Help panels.

F4=VIEW LOG

Description *Display*. Press F4 to view the log.

UEF syntax

PROGRAM description

```
PROGRAM program_name(
  [PASSWORD=password]
  [DESCRIPTION=description]
  [ATTRIBUTES(pgr_attr[,pgr_attr...])]
  [SQL_OWNER_NAME=sql_owner_name]
  [SQL_MODULE_NAME=sql_module_name]
  [UNDEFINED_VARIABLES=undefined_vars]
  LINES(
      [line]...
  )
```

SCREEN description

```
SCREEN screen_name(
  [LANGUAGE=language]
  [DEVICE(dev_row,dev_col)]
  [DESCRIPTION=description]
  [PASSWORD=password]
  [DOMAIN(dom_row,dom_col)]
  [ATTRIBUTES(map_attr[,map_attr ...])]
  [MASK_CHAR=mask_char]
  [FILL_CHAR=fill_char]
  [FIELD field_name((row,col),size[,field_attr[,field_attr...])]
   [VERTICAL(v_repeats,v_displacement)]
   [HORIZONTAL(h_repeats,h_displacement)]
   [COLOR=color]
   [MASK=mask]
   [VALIDATION(
      [DEFAULT=default_value]
      [LOW_RANGE=low_range]
      [HIGH_RANGE=high_range]
      [VARIABLE=dynamic_valid_variable_name]
      [LIST_ITEMS(item[,item...])]
      [ROUTINE=routine_name]
      )]
  )]...
)
```

NOTE

The IBM environment does not have WINUP and WINDOWN terminal functions.

Internal FILE description

```
FILE file_name(
  [DESCRIPTION=description]
  [VIEW_PASSWORD=view_password]
  [ALTER_PASSWORD=alter_password]
  [INSERT_PASSWORD=insert_password]
  [STATUS=status]
  [RECORD_LAYOUT=layout_name]
  [FIELD field_name(
      TYPE=field_type
      [ATTRIBUTES(field_attr)]
      [DIMENSIONS(dim_left,dim_right)
   ])]...
)
```

Internal file DATA description

```
DATA file_name([file_data,...]
)
```

PROMPTER description

```
PROMPTER prompter_name(
  [LANGUAGE=language]
  [PASSWORD=password]
  [DESCRIPTION=description]
  [NEXT_PROMPTER=next_prompter_name]
  LINES(
[line])...
)
```

External file (ACCESS) description

```
ACCESS access_name(
  [DESCRIPTION=description]
  [VIEW_PASSWORD=view_password]
  [ALTER_PASSWORD=alter_password]
  [INSERT_PASSWORD=insert_password]
  [SHARE_PASSWORD=share_password]
  [STATUS=status]
  [ATTRIBUTES(access_attr[,access_attr ...])]
  [FDL_NAME=fdl_name]
  FILE_NAME=file_name
  FILE_TYPE=file_type
  RECORD_TYPE=record_type
  [MAX_RECORD_LENGTH=max_record_length]
  [REFERENCE_VARIABLE=reference_variable]
  [NUM_REPEATS=controlling_repeat_field
  FIRST_REPEATING_ELEMENT=first_repeat_field]
  [READ_TIMEOUT=mailbox_timeout]
  [KEY_OF_REFERENCE=n]
  [ACCESS_METHOD=file_access_method]
  [SAP_RELEASE=sap_release]
  [SAP_COMP=sap_comp]
  [SAP_TYPE=sap_type]
  [SAP_RECID=sap_recid]
  [AIX_NAME=aix_name]
  [FIELD field_name(
      TYPE=field_type
      FORMAT=format
      [ATTRIBUTES(field_attr)]
      [SIGN=sign]
      [DIMENSIONS(dim_left,dim_right)]
      [DECIMAL_PLACES=decimal_places] POSITION=position
      OFFSET=offset
  )] ...
)
```

ULTRA/TOTAL description

```
ULTRA ultra_name(
  [DESCRIPTION=description]
  [VIEW_PASSWORD=view_password]
  [ALTER_PASSWORD=alter_password]
  [INSERT_PASSWORD=insert_password]
  [STATUS=status]
  FILE_NAME=pdm_data_set
  FILE_TYPE=file_type
  [LINK_PATH=linkpath]
  [RECORD_CODE=record_code]
  [REFERENCE_VARIABLE=ref_var]
  DBMOD=dbmod_name
  [CODE_ELEMENT=code_element]
  [FIELD field_name(
      ELEMENT=pdm-element
      TYPE=field_type
      FORMAT=format
      [ATTRIBUTES(field_attr)]
      [SIGN=sign]
      DIMENSIONS(dim_left,dim_right)
      [DECIMAL_PLACES=decimal_places]
  )]...
)
```

INTERFACE description

```
INTERFACE interface_name
  [DESCRIPTION=description]
  [PASSWORD=password]
  [ATTRIBUTES(attr[,attr ...])]
  [ROUTINE=routine_name]
  [IMAGE=image_name]
  [PIPE_NAME=pipe_name]
  [PIPE_SIZE=pipe_size]
  [TIMEOUT=timeout]
  [PROGRAM=progname]
  [STATUS=status]
  [RECORD_LAYOUT=layout]
      [FIELD field_name(
      TYPE=field_type
      FORMAT=format
      [ATTRIBUTES(attribute)]
      [SIGN=sign]
      DIMENSIONS(left,length)
      [DECIMAL_PLACES=decimal_places]
  )] ...
)
```